All Electron User programs work on BBC Micros with OS 1.2 and Basic II

A Database Publication

electron user

Vol. 1 No. 8 May 1984 £1

Spin the wheels of fortune!
See Page 42

Test your skill at rally driving

Send secret messages with our micro coder

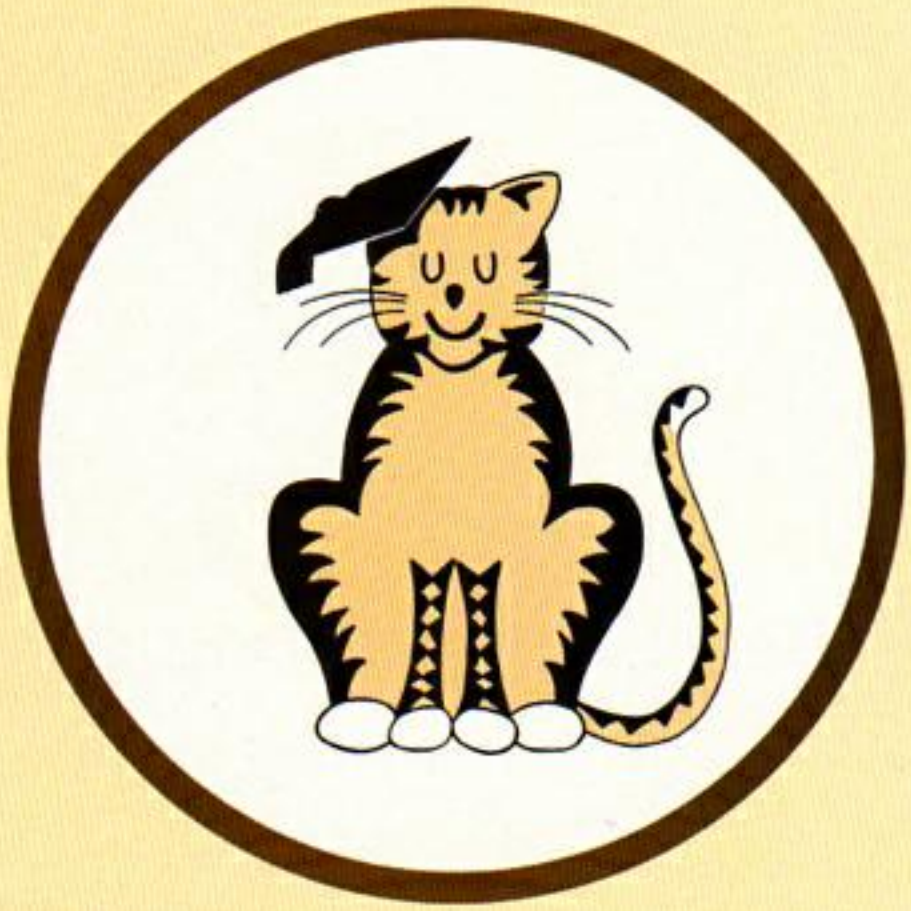
More helpful advice for first-time users

Play your Electron at Tic-Tac-Toe

Win a Joyport joystick interface in our FREE competition

Exciting News for Electron Users
Now Available

CHESHIRE CAT
EDUCATIONAL SERIES
from
AMPALSOFT



CHESHIRE CAT

The First name in Educational Software.

An exciting range of top quality programs covering all needs from pre-school to 'A' level.

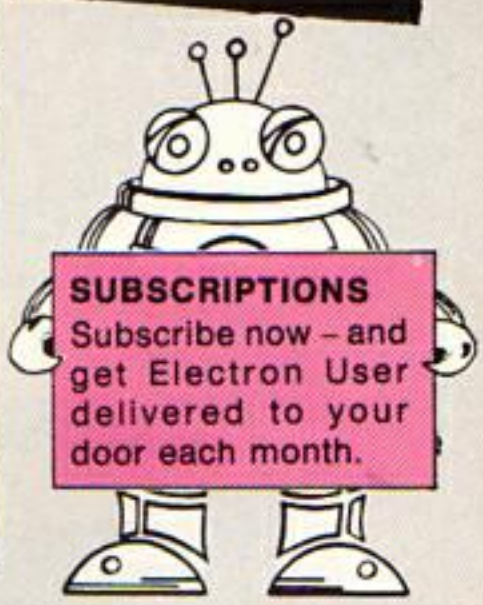
Ampal Computer Services Ltd.
31 Woodbridge Road, Darby Green, Blackwater,
Camberley, Surrey.
Tel: (0252) 876677

Ring 0252 876677
for your nearest stockist

CONTENTS

Vol. 1 No. 8 May 1984

electron user



Managing Editor
Derek Meakin
Features Editor
Pete Bibby
Production Editor
Peter Glover
Layout Design
Heather Sheldrick
Advertisement Manager
John Riding
Advertising Sales
John Snowden
Marketing Manager
Sue Casewell

Published by Database Publications Ltd
Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.
Telephone: 061-456 8383 (Editorial) 061-456 8500 (Advertising)
Subscriptions: 061-480 0171 Telex: 667664 SHARET G. Prestel: 614568383.

Subscription rates for 12 issues, post free:
£12 UK
£13 Eire (IR £16)
£20 Europe
£20 Rest of world (surface)
£40 Rest of world (airmail)

Trade distribution in the UK and overseas: Contact Steve Fletcher, Circulation Manager of Database Publications at the above address, or telephone him on 061-480 4153.
Electron User is an independent publication. Acorn Computers Ltd, manufacturers of the Electron, are not responsible for any of the articles in this issue or for any of the opinions expressed.

Electron User welcomes program listings and articles for publication. Material should be typed or computer-printed, and preferably double-spaced. Program listings should be accompanied by cassette tape or disc. Please enclose a stamped, self-addressed envelope, otherwise the return of material cannot be guaranteed. Contributions accepted for publication will be on an all-rights basis.

"ATTENTION EARTHLINGS...

...Beam us down to A&F Software."

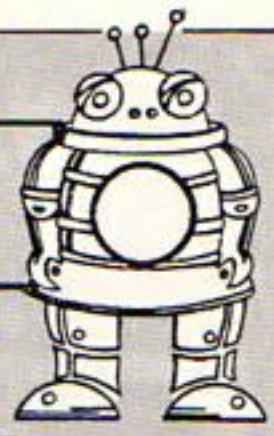
	BBC	ELECTRON	DRAGON	SPECTRUM
CHUCKIE EGG	✓	✓	✓	✓
CYLON ATTACK	✓	✓		
JUNGLE FEVER				✓
PHARAOHS TOMB	✓	✓		✓

A&F Software

Available from W.H. Smiths, Boots John Menzies and all leading computer stores.

Unit 8, Canalside Industrial Estate,
Woodbine Street East, Rochdale, Lancs.
OL16 5LB. Tel: 0706 341111

Electron Eddie-torial



Calling Fred Bloggs!

ONE of the good things about this job is that I get to meet a lot of nice people who are interested in the same sort of things I'm interested in. That's anything to do with the Electron.

I first came across this when I worked on *The Micro User* in the (thankfully) dim and distant past.

An article or letter would come in from someone called Fred Bloggs who I'd never heard of and it would be used in the magazine.

Later another article or a game would turn up, or I'd meet Fred Bloggs at one of our shows and he'd become a friend.

Some regular contributors I've never even met, but they're friends for all that.

Several of these Fred Bloggses have wisely followed me onto *Electron User*, providing articles, inspiration, and a sense of humour.

Mike Cook, Allen Plume and Trevor Roberts, to name but three, came from my *Micro User* days and it's nice to have their support on *Electron User*.

There are, however, quite a few more Fred Bloggs coming to the fore, Electron-using Fred Bloggs who have never written for a magazine before. They usually start their letter: "You probably won't want to use this, but . . .".

One has a penchant for writing programs that move animals across the screen. Another is a school teacher who has become a regular reviewer and promises an article on using the Electron in schools.

And then there's Merlin, our adventures man who just appeared like magic, and another programmer who hails from Fairyland (honest, that's the name of his road).

All were just letters and cassettes on my in-tray at one time. Now they're part of the *Electron User* team.

And every day more contributions from new Fred Bloggs arrive on my desk.

I never know what I'm going to get in the post, or who it's from. There's always something original from someone who I'd never heard of.

It's great fun going through the mail. I'm getting lots of features for *Electron User* and I'm discovering a lot of interesting people.

Your name's not Fred Bloggs, is it?

DYNABYTE
EXPLOSIVE Software

Exciting and original software for the Acorn Electron

NEW! Lemming Syndrome

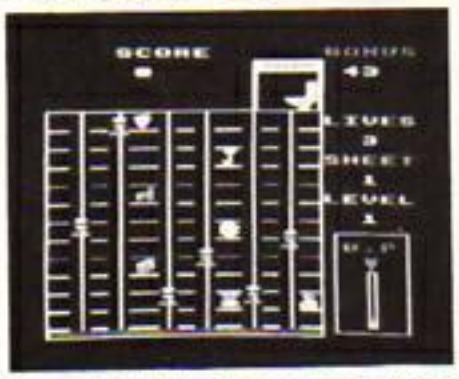
Mad Marco is on the rampage and has blown the bridge to the mainland. The panic-stricken population are hurling themselves into the shark infested waters and your job is to bounce them to safety whilst avoiding the marauding sharks and the desperate attempts of Marco to blow up your liferaft. This highly original, fast and furious game is full of special features and options designed to make your task harder as you get better.

Machine Code **£7.95**

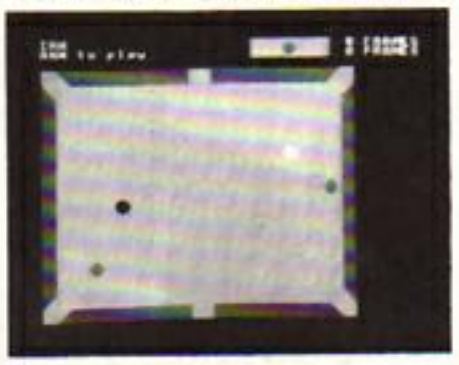
Corporate Climber NEW!

Caught in the capitalistic pursuit of corporate expansion, your ambition is to attain the ultimate accolade – the key to the executive washroom! Avoid the eager taxmen in the lifts ready to hinder your climb to power and beware of too much stress resulting in high blood pressure. **Definitely not for the faint hearted entrepreneur.**

Machine Code **£7.95**



Pool



Classic representation of the real thing incorporating excellent high resolution smooth action graphics for accuracy and making full use of sound. **Start practising now and avoid being hustled.** You control the cue angle and strength of shot. A real pleasure to play.

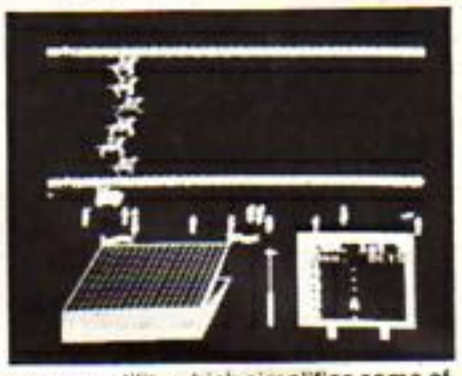
Machine Code **£7.95**

Horserace

An exciting and colourful game complete **in every detail** with tumbling jockeys, realistic horses, TV van, tote and leader boards, waving crowds and much more. Don't lose your money at the track, try HORSERACE instead. Suitable for 1-6 players.

Basic + M/C **£6.95**

(All programs require Series 1.OS)



Also available:
ELECTRON-AID. An extremely useful 2 program utility which simplifies some of the more difficult aspects of programming your computer.
CHARACTER. Easily defines/edits multicoloured characters. VDU 23 statements are automatically generated and can be saved for later use. Characters also displayed normal size on screen.
SOUND LAB. Experiment with up to 7 envelope and 15 sound commands simultaneously. Sounds can be played individually or in sequence. All parameters clearly displayed and easily altered.
Comes complete with full documentation and a user key strip
Excellent value at only **£6.95**

All programs available from most good computer shops or direct from

DYNABYTE SOFTWARE (Dept. EU5)

31, Topcliffe Mews, Wide Lane, Morley, LS27 8UL.

SAE for Catalogue (Please include 50p p&p) Trade Enquiries Phone: 0532-535401

KAY-ESS COMPUTER PRODUCTS

PROFESSIONAL PROGRAMS FOR THE MODEL B AND ELECTRON

SPACE TRAFFIC CONTROLLER—"NEW" (B)(E) **£6.95**
As a space traffic controller you have been stationed at the main robot cargo port of the planet Ore-7. It is your job to get the robot spacecrafts down in one piece. As your confidence increases you can increase the number of crafts allowed within your control area. Warning: not to be played after a hard or hectic day! Pause option.

SPACE TANK (B) **£6.95**
After you SPACE TANK has landed on the planet Orion, a series of alien tanks, surface hoppers, and spacecrafts will attack. How long can you hold out commander? This game makes use of the Beeb's fast scrolling ability. Can be used with either keyboard or joysticks. Top ten table. Pause option.

HORSES (B)(E) **£6.95**
Come on now, don't be shy, choose one of the six horses and let's see what you can do. How many of the fences can you complete at the Orion arena, especially with the clock ticking away? New riders can try one of the more docile horses while others may like to risk one of the more lively beasts! Can be used with either keyboard or joysticks. Top ten table. Pause option.

STAR HAWKS (B)(E) **£5.95**
Can you stop the STAR HAWKS before they stop you? Slow work means the generation of more laser firing mutant hawks. Based on the games of Galaxian and Gorf. Can be used with either keyboard or joysticks. Top eight table. Pause option.

DESIGN (B)(E) **£5.95**
If you like watching your user defined characters run around the screen but are fed up with the time consuming mathematics, then DESIGN is for you! With DESIGN you can draw your characters on an 8x8 grid and let the machine do all the work. DESIGN's features include being able to recall characters for re-editing, displaying VDU 23 commands, and amendable cursor. All characters used in KAY-ESS programs are created using DESIGN.

HANGMAN (B)(E) **£5.95**
Let words become fun again with our three language, (ENGLISH, FRENCH, ITALIAN), version of the popular game of HANGMAN. There are 3 levels of play for each language. All words can be replaced or removed, and new ones can be added. HANGMAN comes with an instruction program giving full details for parents and teachers. Once running prying eyes cannot access the word lists!

★ **CURRENT BEST SELLERS** ★

EARLY YEARS 1 and 2 (B)(E) **£7.95 EACH or £14.00 FOR BOTH**
These two packages are designed to help a young child with some of the simple concepts that they will need in the world. The emphasis is on learning through fun with simple game type tasks to enforce idea's. 'Time tables' are out and Fred the Frog is in! Topics covered include subtraction, addition, recognition, colour, shapes, sizes, sounds/notes, co-ordination, distances, estimates, directions.

EARLY YEARS 1
A) MICKEY THE MONKEY and his apple tree make subtraction fun.
B) COLOUR BLOCKS bring sizes and colours into perspective.
C) MERRY MUSIC turns the keyboard into a musical keyboard.
D) FUNNY FACES presents a line up, which one is the suspect?
E) FRED THE FROG needs co-ordinated help to get across the pond.

EARLY YEARS 2
A) THE POND seems very active today.
B) SPEED is required to keep the cake on the conveyor belt.
C) DIRECTIONS seem to be needed by everyone in Orion village.
D) ORDER the blocks.
E) SID THE SPIDER needs some help to get out of the maze.

For children between 4-8 years of age.

Cheques/P.O.'s should be made payable to KAY-ESS computer products. All prices are fully inclusive. MAIL ORDER ONLY.

KAY-ESS programs are now becoming available at local dealers.

Dealer enquiries welcome.

KAY-ESS computer products previously traded under the name of ORION SOFTWARE.

Available for:
(E) Electron (B) BBC Model B

FREE with all orders is our 3 level version of NOUGHTS AND CROSSES!!!

KAY-ESS Computer Products,
11 Buttercup Close,
Romleighs Park,
Harold Wood,
Essex RM3 0XF.

electron user NEWS

Electron utilities start to pour onto market

ELECTRON software has now left its infancy with the release of a growing number of utilities programs.

The first wave of programs for the Electron consisted almost inevitably of games. These were mostly arcade games but adventures soon followed.

Then, as reported in last month's *Electron User*, educational programs started to head the new releases.

Software companies, already experienced on the BBC Micro, realised that the Electron's potential in the educational field opened up a whole new market.

Now software has entered a third phase, that of the utilities.

Utility programs are neither games programs nor specifically educational. They are designed to make use of the Electron as a tool, rather than a toy or a teaching machine.

From Superior Software of Leeds comes the Electron Disassembler, a utility which allows the user to explore the workings of the Electron's ROM, its operating system and Basic.

The disassembler translates machine code, the Electron's operating language, into the rather more intelligible assembly language.

Another Leeds-based firm, Dynabyte, have produced Electron-Aid, a utility which consists of two programs.

The first, Character, allows the creation and revision of multi-coloured characters.

The second, Soundlab, allows experimentation with the Sound and Envelope commands.

From Salamander Software of Brighton comes the Graphics System, a utility which provides an advanced picture drawing system for Electron users.

This third wave has only just begun. But with software being produced covering such diverse topics as astronomy and personal accounts, it promises to be the most interesting yet.

...AND ADD-ONS ARE ON THE INCREASE, TOO

First Byte's switched joystick interface

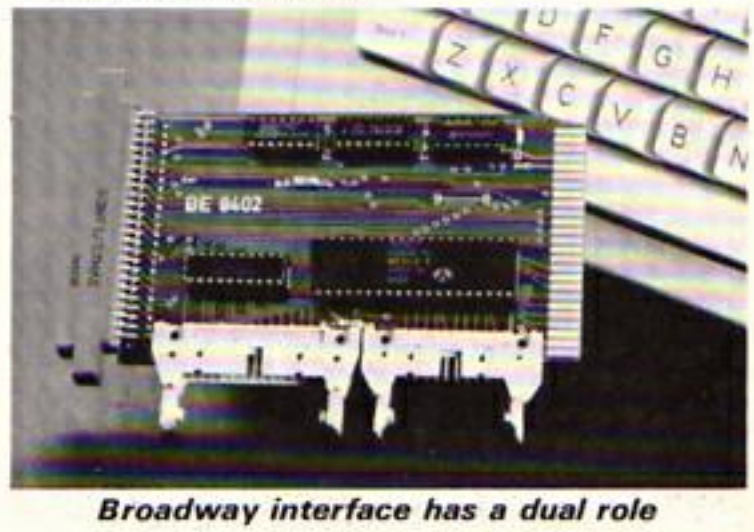
AT ONE time seemingly as elusive as the Electron itself, hardware add-ons are reaching the market in increasing numbers.

Derbyshire based First Byte Computers chose the *Electron and BBC Micro User Show* to release their switched joystick interface.

Capable of taking all standard "Atari-style" joysticks, FBC say that reading the interface is considerably quicker than normal keyboard input or reading an A/D converter.

They have sent pre-production interfaces to all the leading software houses in an effort to

→



Broadway interface has a dual role

Deliveries improve

THE Electron famine appears to be slowly easing. Dealers are reporting that, while they are still not getting all the Electrons they could sell, deliveries are increasing.

Hopes are that by the end of summer the huge backlog will be easing.

Meanwhile Acorn are becoming more open about what has been causing the problems.

Acorn's marketing manager, Tom Hohenberg admitted that a lot of the trouble stemmed from the ULA, the custom made chip at the heart of the Electron.

The world chip shortage made the situation worse.

Things are getting better but Acorn are carefully avoiding giving the numbers of Electrons being produced.

THE DAY A GHOST GOT AN ELECTRONICS UPDATE

THE GHOST of electronics pioneer Sebastian de Ferranti materialised recently – just to get a glimpse of the Electron.

Complete with silver topped cane, tail coat and bowler hat, the apparition of the Victorian gentleman almost brought a northern town to a halt as he dropped into its main micro shop.

It was all for the benefit of a government film unit which had resurrected the 19th century genius in the form of actor John Rankin for a television programme about the micro revolution.

The film, which has been booked by 60 TV stations around the world, aims to highlight the enormous progress made since the days de Ferranti became a pioneer in the large scale use of electricity and brought light into millions of British homes.

Born in Liverpool in 1864, Sebastian de Ferranti invented Britain's first major power station, and the company that still bears his name is now prominent in making micro chips for computers.

The aim of the film crew at Wilmslow Micro Centre was to shoot footage of the ghost examining the Electron – at the heart of which lies a unique chip manufactured by Ferranti at Chadderton.

However, the actor playing the part was first to admit when it came to electronics he couldn't hold a candle to Ferranti himself.

"I'm afraid it's a subject way above my head", 28-year-old John Rankin told *Electron User.*

The film's director did not see this as a disadvantage.

"What we have been trying to capture here is the amazement that would have been felt by de Ferranti at what has been happening in the last 100 years or more", he said.

ADD-ONS BOOM

From Page 7

standardise joystick software.

From Broadway Electronics of Bedford comes a combined printer interface and user port.

Complete with drive software and screen dump routine, the module is claimed to be fully centronics compatible and designed with future expansion in mind.

In the pipeline are a disc interface, joystick controls and sideways ROM board, together with a motherboard for multiple installation.

Meanwhile Acorn are promising their own Electron printer and joystick interface for late May.

BT sign Electron boards contract

ACORN has signed a "cast iron" contract with British Telecom guaranteeing the delivery of several thousand Electron boards by the second half of this year.

They are to be incorporated into the new Merlin Healthnet Workshop which, although still under wraps, is set to be marketed later this year.

Designed to provide an electronic mail link between health centres and local hospitals, the workstation is already generating considerable interest within the health industry.

"We selected the Electron board because of its suitability and price, and the fact it has a real keyboard", a BT spokesman told *Electron User.*

Asked how they could be assured of deliveries while Acorn still has a backlog of more than 200,000 orders for the Electron to be filled, the BT spokesman replied:

"We have an absolutely cast iron contract with Acorn which guarantees us delivery . . ."

No close-down

REPORTS that the Electron production line in Malaysia has been shut down have been strongly denied by an Acorn spokesman.

Contradicting rumours that production difficulties had led to its closure, he said confusion may have arisen because the first Malaysian contract was coming to an end.

"All that has happened is that they have produced the number of Electrons they were under contract to produce", he said.

He declined to tell *Electron User* how many that was, or whether there would be another contract with the Malaysian producers.

Regardez!

NEW ELECTRON PROGRAMS

- ★ Pupils
- ★ Teachers
- ★ Travellers
- ★ Students
- ★ Graduates
- ★ Linguists
- ★ In fact anyone having an interest in French will benefit from this unique language learning aid
- ★ Also available for

BBC model B
SPECTRUM 48K

★ Ready made lessons provide an enormous vocabulary of words, phrases and verbs arranged in subject groups.
★ Lessons can be run in three ways; learning, self-test or speed and accuracy test where you key in the answers.
★ Lesson displays include all French accents, different colours for masculine and feminine words.
★ Full tape editing facilities allow an infinite number of new or updated lessons to be created and stored for later use.

Choice of Level A or B cassettes with totally different vocabularies.
£9.95 each (P&P inc.)
Both cassettes include extensive word lists; verbs and phrases are introduced in Level B. Available from dealers or mail order.
State BBC, Spectrum or Electron
Also available "The German Master" "The Spanish Tutor".

KOSMOS
SOFTWARE
Unit B
1 Pilgrims Close, Harlington, Dunstable, Beds. LU5 6LX
Tel: 05255 3942

EPIC ADVENTURES

FULL-SCALE MACHINE CODE ADVENTURES FOR THE BBC AND ELECTRON

OUR AMAZING NEW ADVENTURE IS NOW AVAILABLE

THE WHEEL OF FORTUNE

They said it couldn't be done on the Beeb – but we've done it!

The Wheel of Fortune is a classic puzzzle adventure, with 250 locations, and brings the following advanced features together for the first time:-

* Sophisticated language and speech interpreters capable of accepting single or multiple commands, up to 254 characters in length. Complex multiple commands are phrased just as you would speak them.
* Moving characters with varying moods. These characters remain active whether you type anything or not. Their reactions to you will depend upon the way in which you have previously treated them. The speech interpreter allows you to talk to them, to either give them commands or information, or to ask them questions.
* Instant half-screen teletext graphics for each location (BBC only). These remain on screen with the text and both may be studied simultaneously. The graphics may be switched on or off, as required.
* You may save your position on tape OR DISC, using a different filename for each position.
* Up to 10 commonly-used command sentences can be stored and called up as required. The stored sentences may be changed during the game.
* No frustrating illogical mazes * Humorous character behaviour * Scoring * Fast response * Fully disc compatible * Etc. Etc.

This masterpiece of programming is available for BBC or Electron (state which) for only **£9.95**

Also available are our 3 popular text adventures. Each has approx. 230 locations and costs just **£7.95**
1) Castle Frankenstein 2) The Quest for the Holy Grail 3) The Kingdom of Klein
P&P FREE if ordering 2 or more games, otherwise add 50p

EPIC SOFTWARE DEPT E4

10 GLADSTONE STREET, KIBWORTH BEAUCHAMP, LEICESTER LE8 0HL

Please make cheques payable to EPIC SOFTWARE
All our programs are available for immediate despatch — Dealer enquiries welcome

ANSWER BACK
SENIOR
QUIZ
GENERAL KNOWLEDGE

THE ULTIMATE EDUCATIONAL QUIZ FOR AGES 12 & OVER

BBC (32K)•ELECTRON

The ANSWER BACK Quiz provides an incredible adventure in education by combining a compelling Space-Age game with an immense series of questions on General Knowledge. The thought-provoking and well-researched quizzes contain an enormous total of 750 questions with 3000 answer options covering the following subjects:

☐ Astronomy ☐ Music ☐ Natural History ☐ Famous People ☐ Science ☐ Sport ☐ History ☐ Art and Architecture ☐ Know your Language ☐ Discoveries and Inventions ☐ Legends and Mythology ☐ Geography ☐ Literature ☐ Films, TV and Theatre ☐ Pot Luck

The highly sophisticated control program rewards each correct answer with another turn in the colourful, animated game.

FEATURES INCLUDE

☐ Multiple choice answers ☐ True or False? ☐ Find the missing letters ☐ "Pass" facility ☐ Immediate correction of errors ☐ Timer option ☐ Performance summary ☐ Re-run of questions passed or incorrectly answered ☐ Full facilities for creating and saving an unlimited number of new quizzes

Available from your computer store or by mail order Price £10.95.

AVAILABLE SHORTLY:
ANSWER BACK
Junior Quiz
for the under 11's

The ANSWER BACK Senior Quiz will educate and fascinate ANYONE over 11 years old.

KOSMOS SOFTWARE, 1 Pilgrims Close, Harlington, DUNSTABLE, Beds. LU5 6LX
Please send me the ANSWER BACK Senior Quiz for the BBC/ELECTRON computer.
Mr/Mrs/Miss.....
Address.....
.....
.....Post code.....
I enclose a cheque/postal order for £10.95 payable to KOSMOS Software

Part Four of PETE BIBBY's introduction to programming

Name your numbers and LET them have some sp

IN the last article we covered strings, collections of letters and symbols that we want the Electron to treat as one lump.

We saw that we could use labels ending in the dollar sign, $, to refer to these strings.

It wasn't all that exciting but we found we could run programs like this:

```
10 REM PROGRAM I
20 LET A$=" GRAIN "
30 LET B$=" DOG "
40 LET C$=" DUCK "
50 PRINT A$;B$;C$
60 PRINT B$;A$;C$
70 PRINT C$;A$;B$
```

Not exactly earth-shattering, but the program does have its important points.

Notice how once I had assigned *A$*, *B$* and *C$* with the LET statements in lines 20, 30 and 40, I was then able to use the labels, or string variables, to print out three different messages.

I saved myself some typing by using the variable names.

So far we've only given labels to strings. You might ask if we can give labels to numbers and the answer is yes, as shown here:

```
10 REM PROGRAM II
20 LET A=3
30 LET B=5
40 PRINT A+B
```

Ignoring the fact that we could do it in our heads, let's look at the principles involved in Program II. Once you've grasped them, programming will become much simpler.

Line 10 is just the REM statement giving the title of the program.

Line 20 uses a LET command to assign a value of 3 to the variable named *A*.

All this means is that when we refer to *A*, as we do later in the program, the Electron will know that we mean the number 3.

Similarly, line 30 gives *B* the value 5. Line 40 now adds the two together. We could of course have just had line 40 as:

```
40 PRINT 3+5
```

and it would work just as well.

The point is that in Program II we used *A* and *B*, two numeric variables. The Electron was quite happy to use the labels rather than the actual numbers in the final addition. It still gave the correct answer.

If we wanted, we could even add the two variables together and refer to the result by another label. Then we could use a PRINT command to display the result.

Program III shows this method in action:

```
10 REM PROGRAM III
20 LET H=330
30 LET M=430
40 LET Z=H+M
50 PRINT Z
```

Line 20 gives the variable *H* the value of 330, and line 30 labels 430 with the name *M*.

What line 40 does is to tell the micro to add together *H* and *M* and give the result the label *Z*. Line 50 then goes on to display Z.

The point to grasp is that we can do calculations like the above sum just using variable names and let the result have a variable name. While this example is ridiculously easy for the Electron, the principles involved will apply throughout your computing career.

Notice that it doesn't matter what values we give to *H* and *M* in Program III. Lines 40 and 50 will still give the correct answer.

Whatever the numbers assigned to the variables in lines 20 and 30, lines 40 and 50 are arranged so that the two figures are added and the result printed out.

Try typing in lines 20 and 30 with different figures in them and you'll see that the program still adds the two numbers together.

The numbers may differ, but the action of the program remains the same.

This use of labels or variable names can save us quite a lot of time and trouble. Have a go at Program IV and you'll see how.

```
10 REM PROGRAM IV
20 LET D=100
30 LET E=200
40 PRINT D+5,D-5,D*2,D/20
50 PRINT E+24,E-16,E*2,E/25
60 PRINT E-D,D-E,E+D,E/D
```

The last three lines of the program give us the results of 12 different calculations using the two variables *D* and *E*.

If we wanted to do the same calculations with two other numbers such as 400 and 800 the only lines we'd have to change would be lines 20 and 30.

We'd just give the labels *D* and *E* the new values. The rest of the program would stay unchanged and give the required results.

The new lines would be:

```
20 LET D=400
30 LET E=800
```

The program is quite powerful. We can assign any two numbers to the variable names in lines 20 and 30 and it will perform the correct calculations.

The Electron will do the same thing, carry out exactly the same operation on different numbers with very little effort.

Just by changing the values of the variables we could perform hundreds of calculations, far faster than we could on paper – and that is the essence of computing.

Now let's change the subject a little and look at what a LET statement actually does.

We've said it gives a label to a number or a string and that we can refer to that string or number by that label. This is true but there's a little more to it than that.

You probably already know your Electron has 32k of memory for you to use.

This can be looked on as an electronic scrap pad. It's here that all your programs are stored in coded form.

We won't bother about the technicalities of memory, it's not needed at this stage.

We will, however, take a look at what the LET command does with the memory.

Suppose we have a line like:

```
10 LET X=3
```

What this does is to tell the Electron to set aside a part of memory to store a number in. It knows that it's a number, not a string, as the name doesn't end in $.

It is to call this reserved part

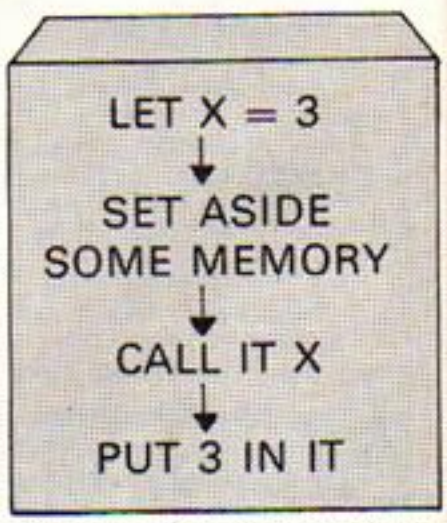
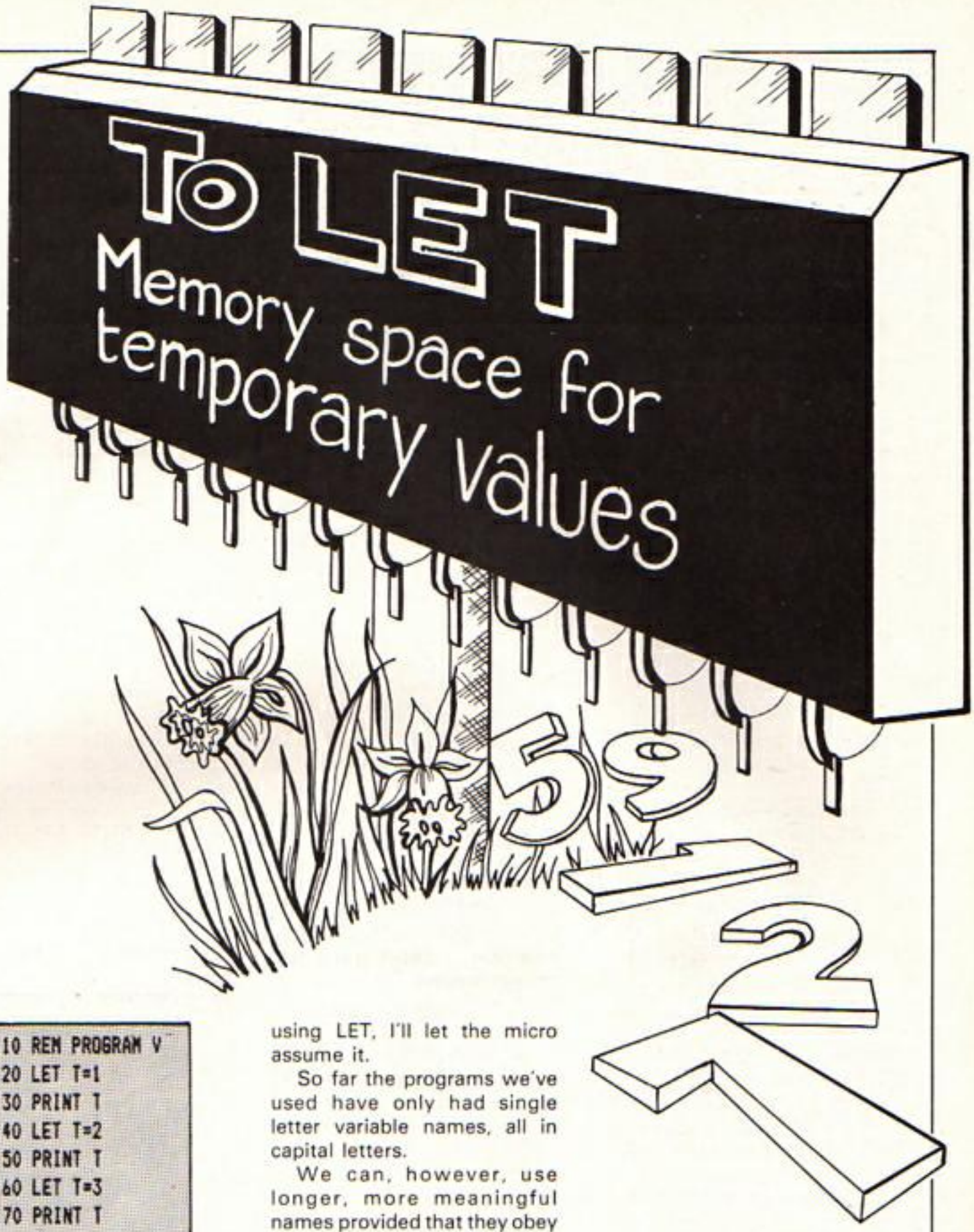


Figure I: Assigning a variable

ace



of memory *X* and it is to put the value 3 into it. Figure I shows this in operation.

Now, when the Electron comes to an *X* in a program it will search through the memory for the part called *X* and use the value it finds stored there.

Should there be no piece of memory labelled *X* it will tell you so with an error message.

If, later on in the program, we have a line such as:

```
200 LET X=7
```

this will cause the Electron to look through its memory for the part called *X* and store the value 7 in it. Now if we have a line such as:

```
210 PRINT X
```

it will print out the value it finds in the part of memory labelled *X*, which is 7.

The old value has gone, the memory only keeps the last value given to that label.

So, to summarise, when we give a number a label, we are setting aside a space in memory, calling that space by the label.

When we later use the label in a program the Electron searches its memory until it finds the part with that label and gives the program whatever value it finds there.

Program V shows this in action. Line 20 sets aside a piece of memory and calls it *T*. Line 30 tells the Electron to display the value it finds in that part of memory labelled *T*.

Line 40 tells the Electron to find the part of memory labelled *T* and put the value of 2 in it.

Line 50 then prints out the value the micro finds in the part of memory labelled *T* which is now the number 2. I leave it to you to find out what lines 60 and 70 do!

```
10 REM PROGRAM V
20 LET T=1
30 PRINT T
40 LET T=2
50 PRINT T
60 LET T=3
70 PRINT T
```

Now that you've done all that typing, I'll let you into a secret about LET. You don't have to use it in the BBC Basic used by the Electron. The Electron will interpret a line such as:

```
10 P=5
```

as:

```
10 LET P=5
```

In both cases, *P* now stands for 5. This means that we could have written Program III as:

```
10 REM PROGRAM III
15 REM (without LET)
20 H=330
30 M=430
40 Z=H+M
50 PRINT Z
```

and the Electron would accept it. From now on I won't be using LET, I'll let the micro assume it.

So far the programs we've used have only had single letter variable names, all in capital letters.

We can, however, use longer, more meaningful names provided that they obey the rules set out in Table I.

Using longer, more appropriate names can really make a difference to understanding how programs work.

Have a look at this:

```
10 REM PROGRAM VI
20 W=10
30 H=20
40 A=H*W
50 PRINT A
```

This prints out the area of a rectangle of width 10 and height 20. Program VII does exactly the same thing but it is much more easily understood from its listing:

```
10 REM PROGRAM VII
20 width=10
30 height=20
40 area=width*height
50 PRINT area
```

You'll notice I have used meaningful variable names and that they are in lower case letters. The names are in small letters for two reasons.

The first is so that there is no inadvertent clash between a variable name and a Basic keyword, of which more later.

Since Basic keywords must always be in capital letters, so using lower case variable names avoids this problem.

The second reason is that the variable names stand out in the listings, separated from the Basic keywords which have capital letters.

It may not help the Electron, but it does help you and

→

From Page 11

anyone who may be reading your listings.

Let's take a brief look at the rules for variable names shown in Table I.

The first says that there must not be any spaces in the name. If you decide to use a variable name with a space in the middle, you'll get an error message.

If you must have a gap, then use the underline character which you'll find on the same key as the down cursor.

And don't use the hyphen instead of the underline. You aren't allowed to use punctuation marks or mathematical symbols in variable names. Nor can they start with a number.

Finally, as we said above, a variable name can't begin with a Basic keyword. A variable *LETTER* would cause the Electron confusion with the Basic keyword LET. It would be better to use *letter*.

RULE	WRONG	RIGHT
No spaces in variable name	sleeping dogs = 3	sleeping_dogs = 3
Must not start with number	2nd time = 35	secondtime = 35
No punctuation marks in name	peter's = 9	peters = 9
No arithmetic operators included in name	night+day = 24	nightandday = 24
Must not begin with a Basic keyword	LETTER$ = "a"	letter$ = "a"

Table I: Rules for naming variables

It seems like a lot of rules at first, but they'll soon become second nature, and the Electron will always tell you when you've got it wrong.

Using meaningful names really helps you to better programming and it's a habit worth getting into.

And that's it for this month. We've covered giving labels to numbers (numeric variables) and had a closer look at what LET does.

We've seen that we don't have to type in LET – the Electron will assume it.

Finally we've learnt the rules for naming variables, both numeric and string.

Next month we'll be looking at how to give values to variables while the program is still running.

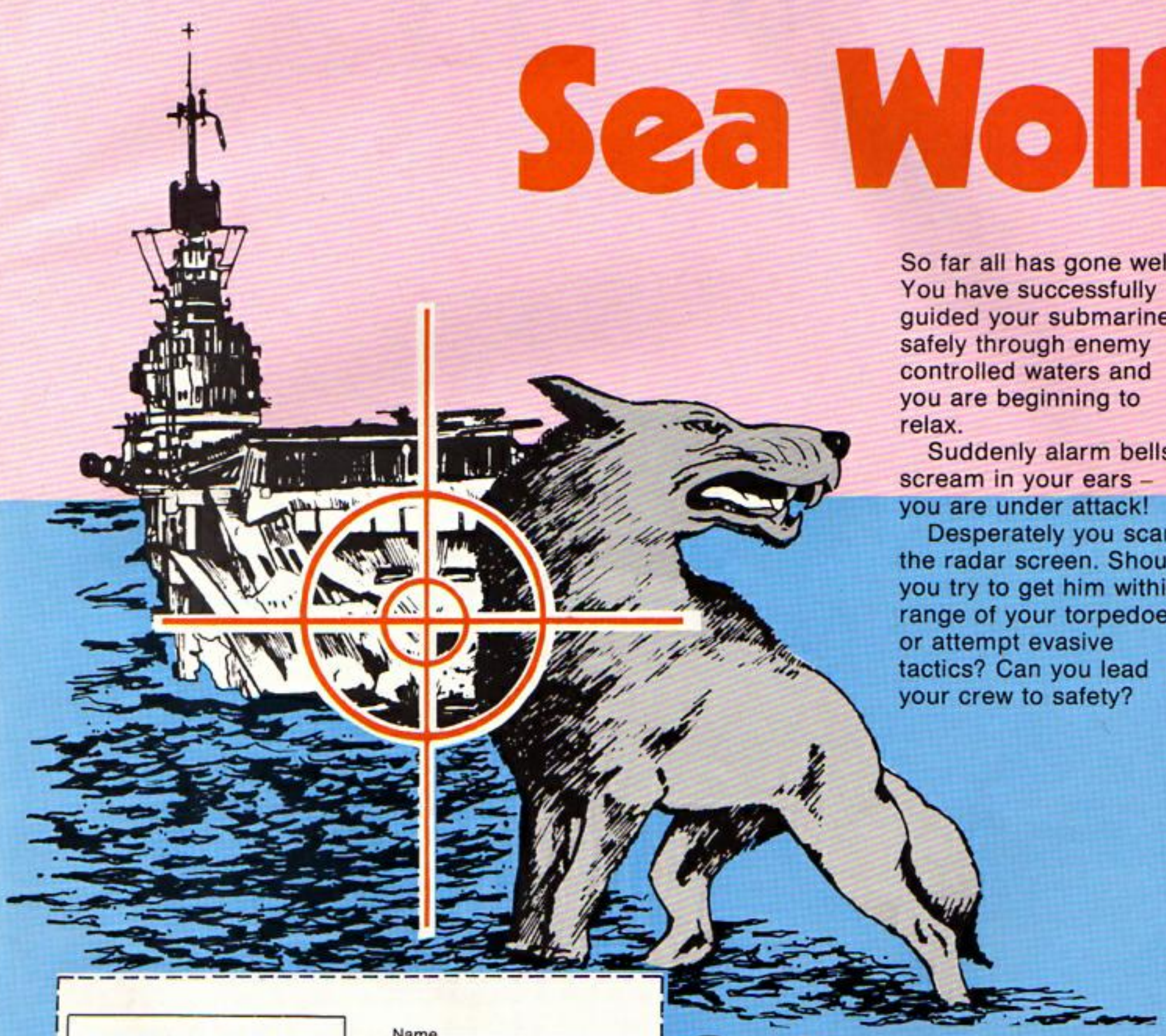
Until then have a look at Program VIII. Can you guess what value *total* will have when it's displayed by the PRINT command of the last line?

Do you understand what's happening?

```
10 REM PROGRAM VIII
20 total=1
30 total=total+1
40 total=total+total
50 total=total+1
60 PRINT total
```

Optima Software

Sea Wolf



So far all has gone well. You have successfully guided your submarine safely through enemy controlled waters and you are beginning to relax.

Suddenly alarm bells scream in your ears – you are under attack!

Desperately you scan the radar screen. Should you try to get him within range of your torpedoes, or attempt evasive tactics? Can you lead your crew to safety?

ORDER FORM

Please send me SEA WOLF:

☐ *BBC 'B' cassette – £6.95*
☐ *Electron cassette – £6.95*

Add 50p p&p
(post free 2 or more)

Name .....

Address .....

.....

.....

.....

.....

☐ I enclose cheque made payable to Optima Software Ltd.

☐ I wish to pay by ☐ Access ☐ Visa

No.....

Expiry date .....

Optima Software Ltd., 36 St. Petersgate, Stockport SK1 1HL.

Software. With a touch of brilliance

NIGEL PETERS investigates the use of graphics windows to enhance your program displays

Send your message through a window!

THIS month we'll be having a close look at Message, a program that Andrew Waite tells me he used to send a greetings message to his uncle.

It's a very simple but also very effective program as you'll see if you type in the listing and run it.

The secret lies in its use of something called the graphics window – a piece of the screen set aside for graphics displays.

The only thing you have to know to understand Message is that you define a graphics window with a VDU24 command.

Of course you have to be in a graphics mode to do it – it won't work in Modes 3 and 6.

This VDU24 is followed by the coordinates of the bottom left hand corner of the graphics window, then those of the top left hand corner.

To get the coordinates you must know that the TV screen is divided into a lot of imaginary points. There are 1280 of them going from left to right and 1024 from bottom to top, as you'll see in Figure I.

You can refer to any point on the screen using two coordinates. The bottom left of the screen is 0,0, and the top left is 0,1023.

The top right is 1279,1023 and the bottom right is 0, 1279.

Usually the graphics screen fills the whole of the screen, but we can change this with the VDU 24.

To achieve the graphics screen that I've coloured red in Figure II all we do is put the Electron in a graphics mode, say Mode 5, type in:

```
VDU 24,40;40;1239;983;
```

and press Return. Don't forget the semi-colons, they're vital.

Nothing much appears to happen, but let's type in:

```
GCOL 0,129
```

and press Return.

Now we'll use CLG to clear the graphics screen we've defined with our original VDU 24 and see what happens. Type in:

```
CLG
```

and press Return. We get a red rectangle.

This is our graphics window. The GCOL changed the background colour to red and when we cleared the graphics window with CLG the window turned to red.

Now let's define another graphics window just inside the first using:

```
VDU 24,80;80;1199;943;
```

and change the background colour to yellow using:

```
GCOL 0,130
```

Now entering CLG will

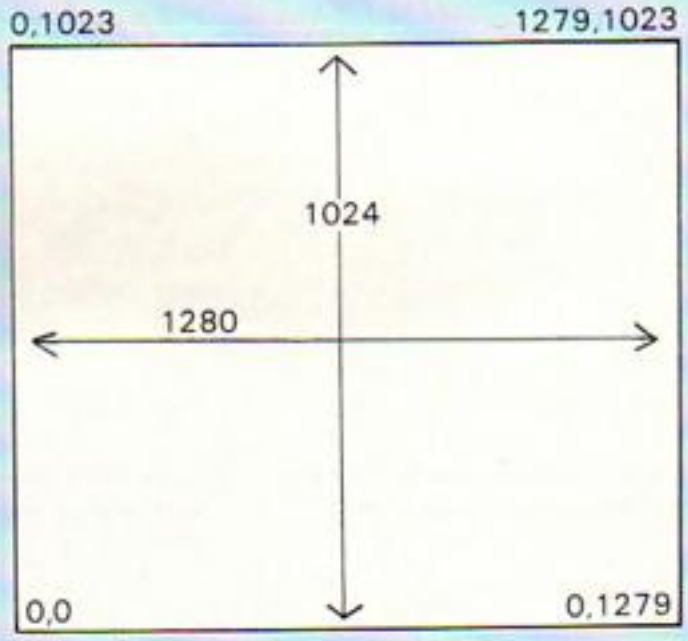


Figure I

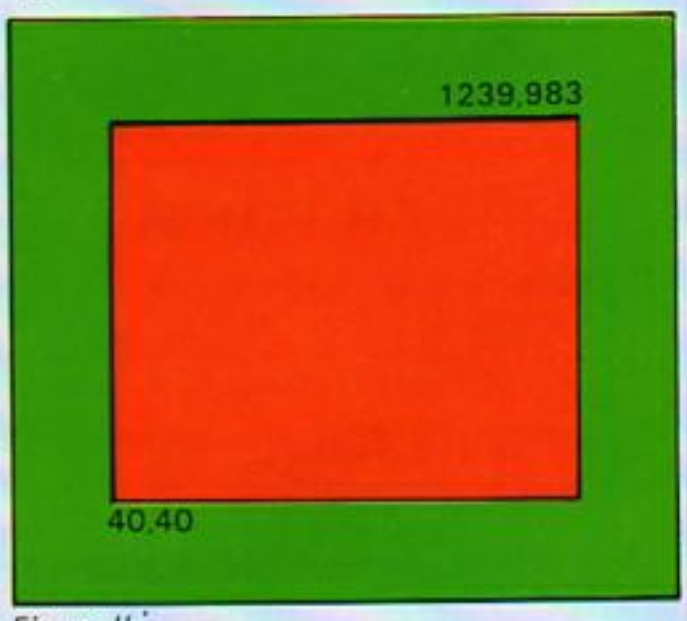


Figure II

produce a yellow rectangle just inside the first.

We've cleared the new graphics window to the new background colour yellow. But it has left the bits outside the new window in the old colour red. Figure III shows what has happened.

Have a go at producing a few different graphics windows. It's a great way of producing fast rectangles and is the technique that Andrew has used to send his message.

The first two lines of the program are REM statements that give information to us humans but not to the Electron.

Lines 30 and 240 form a REPEAT... UNTIL loop that runs the lines that come between them over and over again, endlessly.

Line 230 puts a kind of break on this, holding up the program until a key is pressed. This just allows you to see the message again and again.

Line 40 puts the Electron in Mode 1, as you might have guessed. This is a four colour graphics mode.

The VDU23 on the next line just switches off the flashing cursor.

Lines 60 to 100 set up the five main variables of the program, while line 110 does the main work.

This defines a graphics window.

Where the window actually is depends of the value of *A*, *B*, and *C* when the program executes this line. It does this more than once, as we'll see later.

Line 120 then uses the value that it finds in the variable *D* to alter the background colour of the graphics window and the next line clears the new window to that colour.

Line 140 just makes a beep every time the program comes to it, the pitch depending on the value of the variable *E*.

Lines 150 to 170 alter the values placed in the variables that we've previously used to define the graphics window.

This has the effect of moving the window inwards next time it is defined – see Figure IV.

Line 180 alters the variable that decides the background colour, making sure that it always contrasts with the previous colour.

Line 190 increases the value of *E*.

Line 200 is a powerful one. When the program is run, it sets up a graphics window, then alters all the variables, and then comes to line 200.

If the value of *A* is less than 760 then the program has to go back to line 110 and repeat the whole process over again with the newly altered variables.

This has the effect of displaying a new graphics window and changing the variables again.

If *A* is still less than 760 when the program gets to line 200, it goes back to line 110 and starts all over again creating yet another graphics window inside the others.

That's how we produce all those nice boxes on the screen.

When *A* is equal to 760, or greater than it, the program doesn't have to go back to line 110 as the condition of the GOTO has been fulfilled.

It then goes on to obey lines 210 and 220 and print the message in the centre of the screen. We've congratulated Andrew on his program. You could, if you wish, insert your own messages.

And that's it. Simple, when you know how. Yet very effective indeed. Nice one, Andrew.

```
 10 REM BY ANDREW WAITE
 20 REM (C) ELECTRON USER
 30 REPEAT
 40 MODE 1
 50 VDU 23,1,0;0;0;0;

 60 A=0
 70 B=1279
 80 C=1023
 90 D=129
100 E=0
110 VDU 24,A;A;B;C;
120 GCOL 0,D
130 CLG
140 SOUND 1,-15,E,1
150 A=A+20
160 B=B-20
170 C=C-20
180 D=D+2
190 E=E+40
200 IF A<760
    THEN GOTO 110
210 PRINT TAB(15,15);"
    NICE ONE "
220 PRINT TAB(15,16);"
     ANDREW  "
230 WAIT$=GET$
240 UNTIL FALSE
```

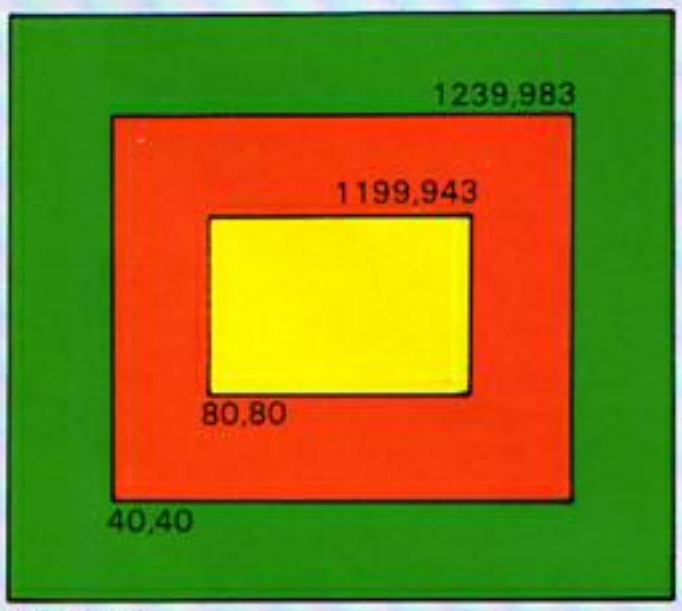


Figure III

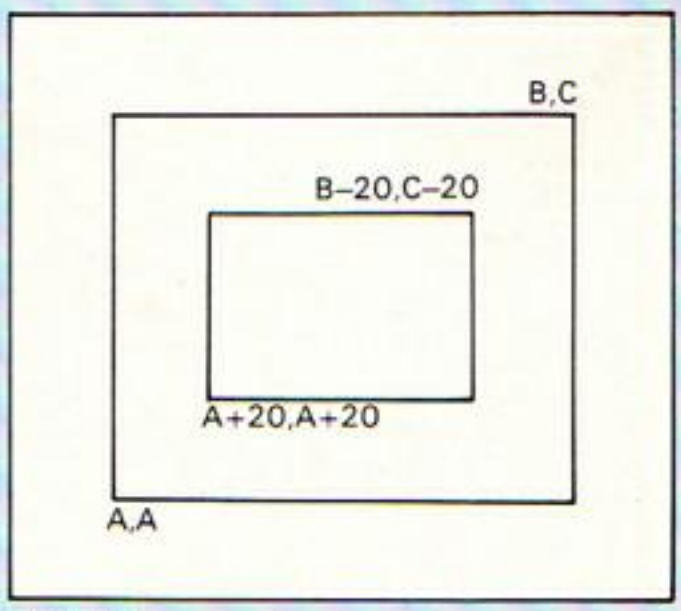


Figure IV

National Micro

1 year GUARANTEE on all our products

BARGAIN OF THE MONTH

WORD HANG (BES)
An educational version of the popular hangman game.
Special price valid to May 31, 1984 ..... **£7.97**

ALL PRICES INCLUDE VAT

How to get a generous discount on all your purchases!

National Micro Centres Computer Club

WHAT YOU DO

Either purchase three items of software for the price of two (lowest priced item free) or make purchases to the value of £100 or more.

WHAT YOU SAVE

1. All subsequent purchases totalling over £25 are eligible for discount* of:
   ● *10% on software* ● *5% on hardware*
   when you quote your membership number.
2. If you buy at least one item of software every month you get 10% discount on all such purchases. Ask for a registration card to take advantage of this scheme.
3. The opportunity to enter the members' competitions run throughout the year. Just ask for competition entry form when you send in your order.

Ask for an application form at any of our outlets or write for a form to:

National Micro Centres Computer Club,
36-38 St Petersgate,
Stockport SK1 1LH.

** Discount only applies to our normal prices and not to any special offer prices which may be in force for the time being.*

Our top best-sellers

Birds of Prey (Romik)
A fast moving invaders type game where the aliens in space take the form of birds. Good value for money. **£6.99**

Twin Kingdom Valley (Bug-Byte)
Not only a good adventure but all 175 locations are drawn in full-screen hi-res graphics. A sophisticated adventure game. **£9.50**

Chuckie Egg (A & F)
Just when you thought it was safe to go back on the farm this game makes you think again. The idea is to collect eggs before storks devour corn or you. A progressive game requiring extremely high skill levels. The nightmare has begun! **£7.90**

Killer Gorilla (Micropower)
Fast becoming a cult game. Dodge tumbling barrels and blazing fireballs as you battle to rescue the damsel in distress. Gripping multi-level action. **£7.95**

Cylon Attack (A & F)
"Outstanding . . . quite simply excellent . . . the graphics leave most other games standing". – Micro User. **£7.90**

GAMES

A & F
Cylon Attack **£7.90**
Chuckie Egg **£7.90**

Acornsoft
Draughts **£9.20**
Meteors **£9.20**
Snapper **£9.20**
Starship Command **£9.20**
Chess **£9.20**
Forth **£16.10**
Graphs & Charts **£9.20**
Lisp **£16.10**
Monsters **£9.20**

Alligata
Bugblaster **£7.95**
Lunar Rescue **£7.95**
Fruit Machine **£5.95**

Bug-Byte
Twin Kingdom Valley **£9.50**

Dr. Soft
747 Simulator **£7.95**

Digital Fantasia
10 Little Indians **£10.29**
Arrow of Death Part 1 ..... **£10.29**
Arrow of Death Part 2 ..... **£10.29**
Circus **£10.29**
Escape From Pulsar 7 ..... **£10.29**
Feasibility Experiment **£10.29**
Golden Baton **£10.29**
Persus Andromeda **£10.29**
Time Machine **£10.29**
Wizard of Akyrz **£10.29**

Ivan Berg
. . . I Do **£12.65**
Crime and Detection Quiz **£12.65**
History Quiz **£12.65**
Music Quiz **£12.65**
Royalty Quiz **£12.65**
Science Fiction Quiz **£12.65**
The Dating Game **£12.65**
Theatre Quiz **£12.65**

Micropower
Adventure **£7.95**
Bandits at 3 o'Clock **£7.95**
Croaker **£7.95**
Cybertron Mission **£7.95**
Danger UXB **£7.95**
Escape from Moonbase Alpha **£7.95**
Felix & The Fruit Monster **£7.95**
Felix In The Factory **£7.95**
Galactic Commander **£7.95**
Intergalactic Trader **£7.95**
Killer Gorilla **£7.95**
Moon Raider **£7.95**
Nemesis **£7.95**
Positron **£7.95**
Swoop **£7.95**

Optima
Seawolf **£6.95**
Bed Bugs **£6.95**

Romik
Alien Break-In **£7.95**
Atom Smasher **£7.95**
Birds Of Prey **£6.99**

Superior Software
Centibug **£7.95**
Alien Dropout **£7.95**
Invaders **£7.95**
World Geography **£7.95**
Fruit Machine **£7.95**
Constellation **£7.95**

Centres

MAIL ORDER DIVISION

Phone your order on **061-429 8080**

Answering service outside normal office hours
Or use the order form below

DATA STORAGE

A data cassette recorder that perfectly matches the Electron

£42.95

Fully compatible with the Electron and specifically designed for data saving and loading, the Pye data cassette recorder (with free power pack and Electron lead) is guaranteed to end your data storage and retrieval problems.

HOME & BUSINESS

Acornsoft
Personal Money Management **£11.50**
Optima
Datext **£9.95**

UTILITIES

Optima
Optimon **£9.95**
Superior Software
Disassembler **£7.95**

EDUCATIONAL

Acornsoft
Creative Graphics **£9.20**
Graphs & Charts **£9.20**
Tree of Knowledge **£9.20**
Alligata
Primary Art **£7.95**
B.E.S.
Animal Vegetable Mineral **£8.97**
Happy Letters **£8.97**
Happy Numbers **£8.97**
Map Rally **£8.97**
Timeman One **£8.97**
Timeman Two **£8.97**
World Wise **£8.97**
Golem
Education 1 ..... **£8.00**
Education 2 ..... **£8.00**
Fun With Words **£8.00**
Super Life **£4.95**
Mirrorsoft
1st Steps with the Mr. Men **£8.95**
Quick Thinking **£6.95**

Personal shoppers can obtain the products advertised on this page from the following retail stores:

Stockport Micro Centre
4/6 Brown St., Stockport,
Greater Manchester.
Tel: 061-480 0539

Wilmslow Micro Centre,
62 Grove Street,
Wilmslow, Cheshire.
Tel: 0625 530891

JUST ARRIVED!

Now you can use a joystick with your Electron with First Byte's switched joystick interface. We expect it to be the most asked-for add-on of them all. This plug-in cartridge takes standard Atari-style joysticks, which are far more popular – and cheaper – than analogue joysticks. Lots of joystick games are now being produced for the Electron. So be one of the first to add this exciting new dimension to your games playing!

First Byte joystick interface **£24.95**
Kempston joystick **£11.74**
Sureshot joystick **£15.05**

MONITORS

Microvitec 1441 Hi (Metal) **£506.00**
Microvitec 1431 Low (Metal) **£247.25**
Microvitec 1451 Med (Metal) **£373.75**
Kaga K12G Green **£125.35**
Kaga K12A Amber **£136.85**
Kaga K12R1 Low Res **£274.85**
Kaga K12R2 Med Res **£327.75**
Kaga K12R3 Hi Res **£458.85**
Nordmende 14" TV/Mon **£250.70**
Nordmende 14" TV/Mon Remote **£264.50**
Zenith 12" Green **£86.25**

ALL PRICES INCLUDE VAT

Be among the first to own the electrifying Electron!

The baby brother of the BBC Micro has been hailed by the computer press as a big breakthrough in power and price. This exceptional machine is already being forecast to be the top-selling micro of 1984. It comes complete with an introductory cassette of 15 programs, a very comprehensive user guide and an easy to understand DIY book on programming.

£199.00

Stocks are limited, so it's very much a case of first come first served. We promise that no cheques will be cashed until the machine is despatched.

System Sac **£19.95**
Electron Dust Cover **£2.95**

ORDER FORM

Post to:
NATIONAL MICRO CENTRES,
36 St. Petersgate,
Stockport SK1 1HL

Item	*Please supply the following:* Qty	Total £	p
.....	.....	.....	
.....	.....	.....	
.....	.....	.....	
.....	.....	.....	
.....	.....	.....	
.....	.....		
		.....	

Attractive credit terms
Phone for details

Carriage ..... FREE.....
TOTAL .....

Please indicate method of payment:

☐ Cheque payable to National Micro Centres

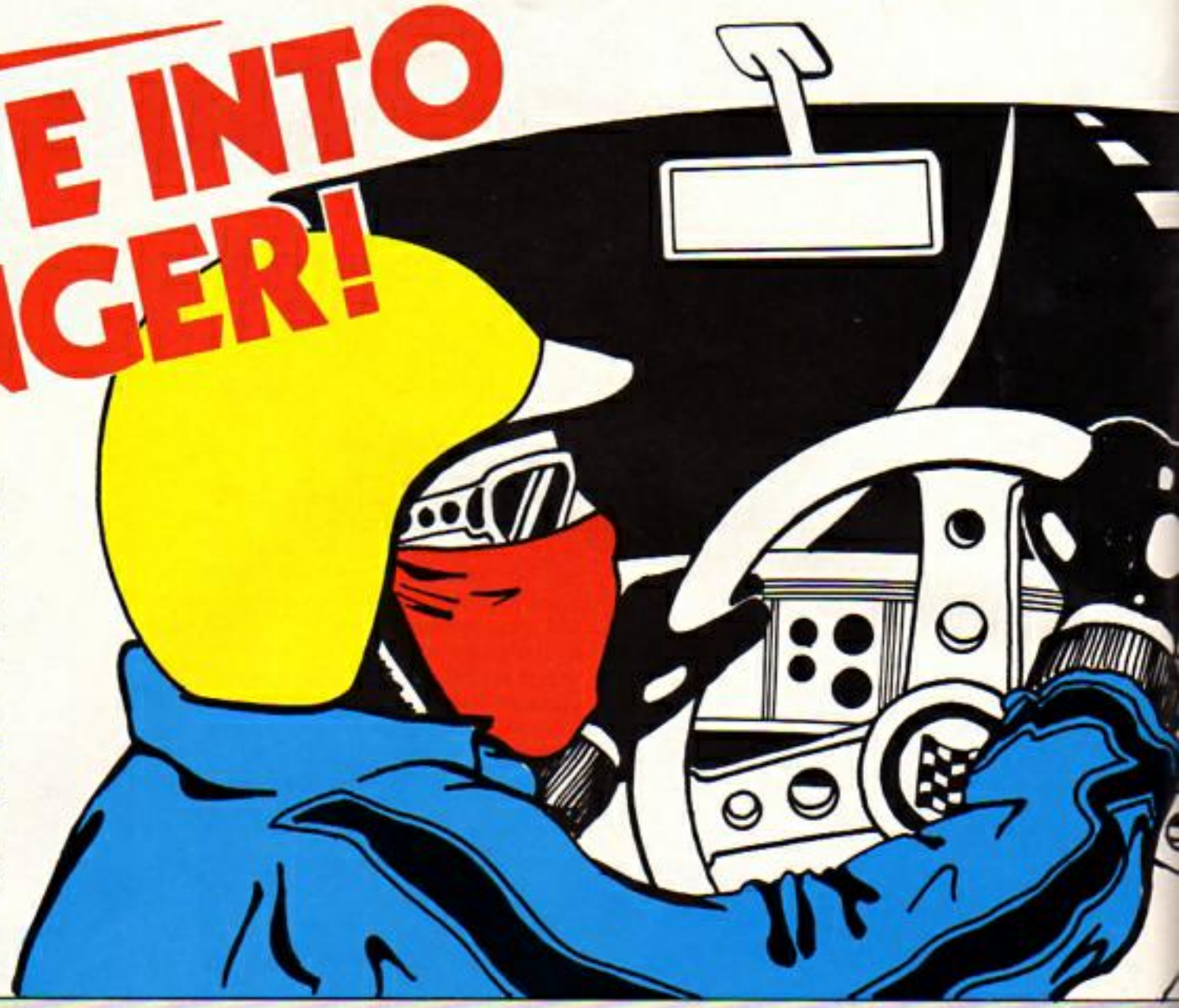
☐ Access/Barclaycard No.

☐

Name .....
Address .....
.....
.....
Tel. No. .....
Signed .....

Enjoy the thrills and spills of this exciting rally driving game by ERIC

DRIVE INTO DANGER!



HAVE you ever fancied having a go at being a rally driver? Well, with Rally Driver you can turn your Electron into a driving simulator and try to steer your way round a course in the fastest time possible.

It's just like the real thing. You can steer from left to right, accelerate, brake and – if you're not careful – crash.

All the instructions are in the program. The rest is up to you.

Drive carefully!

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
10 REM RALLY DRIVE (C)
   ELECTRON USER by
   Eric H. Crisp
20 MODE 6
  :PROCInstruct
  :MODE 5
30 REPEAT
40 PROCInitial
50 REPEAT
60 PROCRoad
  :PROCKeys
  :PROCTest
70 UNTIL F%
80 MODE 6
  :PROCResult
  :MODE 5
90 UNTIL FALSE
100 DEF PROCCalc
110 U%(P%)=(CP%-X%(P%))/(P%
    +1)
120 L%(P%)=U%(P%)-2*(896-V%
    (P%))
130 R%(P%)=U%(P%)+2*(896-V%
    (P%))
140 ENDPROC
200 DEF PROCDraw
210 GCOL 3,3
  :V%=V%(P%)
  :W%=V%(P%+1)
220 VDU 25,4,U%(P%);V%;25
    ,5,(U%(P%+1)+U%(P%))
    DIV 2;(V%+W%)DIV 2;
240 VDU 25,4,R%(P%);V%;25
    ,5,R%(P%+1);W%;25
    ,4,L%(P%);V%;25,5
    ,L%(P%+1);W%;
250 ENDPROC
300 DEF PROCInitial
310 CS%=0
  :CY%=0
  :CP%=320
  :C%=0
  :L%=0
  :PP%=0
  :D%=0
  :F%=0
320 FOR P%=0TO 6
  :X%(P%)=0
  :V%(P%)=896-640
    DIV (P%+1)
  :PROCCalc
  :NEXT
330 TIME =0
  :COLOUR 129
  :@%=6
  :VDU 29,640;160;23;8202
  ;0;0;0;
340 FOR P%=0TO 4
  :PROCDraw
  :NEXT
350 VDU 18,0,1,25,4,-640;-1
    60;25,4,-640;160;25
    ,85,640;-160;25,85
    ,640;160;
360 VDU 18,0,2,25,4,560;210
    ;25,85,640;-4;25,85
    ,530;224;25,85,530;20;2
    5,85,500;234;25,85
    ,470;224;25,85,530;20;2
    5,85,440;200;
370 VDU 25,85,320;44;25
    ,85,320;224;25,85
    ,160;64;25,85,160;244;2
    5,85,0;76;25,85,0;256;2
    5,85,-160;64;25,85
    ,-160;244;25,85,-320;44
    ;25,85,-320;224;25
    ,85,-530;20;25,85
    ,-440;200;
380 VDU 25,85,-640;-4;25
    ,85,-470;224;25,85
    ,-640;160;25,85,-500;23
    4;25,85,-560;210;25
    ,85,-530;224;18,0
    ,0,25,4,-530;20;25
    ,29,-440;192;25,4
    ,530;20;25,29,440;192;
390 COLOUR 0
  :PRINT TAB(3,27)"SPEED"
    TAB(13,27)"TIME"
  :COLOUR 3
  :ENDPROC
400 DEF PROCInstruct
410 DIM X%(10),V%(10)
    ,U%(10),L%(10),R%(10)
420 PRINT TAB(13,2)"RALLY
    DRIVER"TAB(13,3)"******
    ********"
430 PRINT TAB(6,5)"You
    are on a timed section
    of a""rally. You
    can incur penalty
    points for driving
    on the verge or the
    wrong side of the
    road. A crash scores
```

H. CRISP

PROCEDURES

Line	Proc	
100	Calc	Calculates the screen positions of the road sections on the screen.
200	Draw	Draws a road section.
300	Initial	Initialises variables and the screen for a new game.
400	Instruct	Dimensions arrays and displays the instructions.
500	Keys	Reads the keyboard and adjusts variables accordingly, makes the sound and displays the time and speed.
600	Result	Displays the result at the end.
700	Road	Displays the road in the new position.
800	Test	Tests for position on the road and displays the relevant comment.

CHANGES

To extend the game, line 810 can be changed by increasing the value with which D% is compared, so making a longer rally.

The curvature of the bends can be increased by increasing the random value assigned to C% at the end of line 750. To keep left and right bends equal, the first number should be twice the second number.

The length of the bends can be increased by increasing the random value assigned to L%.

VARIABLES

C%	Curve sharpness.
CP%	Car's absolute position, horizontally.
CS%	Car's steering speed.
CY%	Car's forward speed.
D%	Distance travelled.
F%	Finished flag.
L%	Length of curve.
P%	Road section counter – proportional to distance in front of car.
PP%	Penalty points.
T%	Time delay.
V%	A particular V%(P%) value.
W%	A particular V%(P%) value.

ARRAYS

L%(P%)	Screen x coordinate of section P% of the left verge.
R%(P%)	Screen x coordinate of section P% of the right verge.
U%(P%)	Screen x coordinate of section P% of the white line.
V%(P%)	Screen y coordinate of section P% of the road.
X%(P%)	Absolute x coordinate of section P% of the white line.

```
    nothing."
440 PRINT TAB(6,12)"The
    controls are as follow
    s,"''SPC (12)"A ....
    Accelerator"SPC (22)
    "Z .... Brake"SPC (28)
    "< .... Left"SPC (29)
    "> .... Right"
450 PRINT TAB(8,20)"PRESS
    SPACE TO DRIVE OFF"
460 *FX15,0
470 REPEAT UNTIL GET =32
480 ENDPROC
500 DEF PROCKeys
510 CYX=CYX-4000*INKEY (-66
    )+7000*INKEY (-98)-500
520 IF CYX<500
    THEN CYX=500
    ELSE IF CYX>40000
    THEN CYX=40000
530 TX=TIME +200-SQR (CYX)
    :DX=DX+1
    :SOUND 17,1,CYX
    DIV 800,255
    :ENVELOPE 1,40020
    DIV (CYX+20),4,-2
    ,0,1,2,1,126,0,0,-126
    ,126,0
540 REPEAT
550 CSX=CSX-INKEY (-103)+
    INKEY (-104)
    :PRINT TAB(0,28)CYX
    DIV 500TAB(10,28)
    TIME DIV 100
560 IF CSX>10
    THEN CSX=10
    ELSE IF CSX<-10
    THEN CSX=-10
570 UNTIL TX<TIME
580 CPX=CPX+(CYX*CSX)
    DIV 4000
590 ENDPROC
600 DEF PROCResult
610 IF FX=1
    THEN PRINT TAB(7,5)
    "You travelled ";DX
    DIV 10;".";DXMOD 10;
    " miles before""you
     CRASHED!""
     You also managed to
     incur ";PPX;""penalty
     points."
620 IF FX=2
    THEN PRINT TAB(5,5)
    "WELL DONE! You took
     ";TIME DIV 100;" secon
    ds""but incurred ";PPX
    ;" penalty points."''
    "     Your score is
     ";1000-PPX-TIME
    DIV 50
630 PRINT TAB(5,20)"PRESS
     SPACE TO DRIVE AGAIN"
640 *FX15,0
650 REPEAT UNTIL GET =32
660 ENDPROC
700 DEF PROCRoad
710 PX=0
    :PROCDraw
    :XX(PX)=XX(PX+1)
    :PROCCalc
    :PX=1
720 REPEAT
730 IF PX<>5
    THEN PROCDraw
740 XX(PX)=XX(PX+1)
    :PROCCalc
    :PX=PX-1
    :PROCDraw
    :PX=PX+2
750 IF PX=5
    THEN LX=LX-1
    :XX(6)=XX(5)+CX
    :IF LX<=0
    THEN LX=RND(20)
    :CX=(RND(201)-101)
760 UNTIL PX>5
770 ENDPROC
800 DEF PROCTest
810 IF DX>100
    THEN FX=2
    :ENDPROC
820 IF UX(0)>1600OR UX(0)<-
    1600
    THEN FX=1
    :PPX=PPX+100
    :SOUND 16,-15,4,50
    :FOR PX=0TO 500
    :VDU 19,1,PX;0;19
    ,2,PX+2;0;
    :NEXT
    :ENDPROC
830 IF UX(0)>960OR UX(0)<-9
    60
    THEN PPX=PPX+15
    :PRINT TAB(4,30)"ON
     THE VERGE";
    :ENDPROC
840 IF UX(0)<320
    THEN PPX=PPX+5
    :PRINT TAB(4,30)" WRONG
     SIDE ";
    :ENDPROC
850 PRINT TAB(4,30)"
              ";
    :ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 47

Notebook Part 4

Figure it out

MANDALA is an elementary but very effective program that draws a pattern of fine lines on the screen of your Electron.

The program itself is simple, with only nine active lines. But the logic behind it isn't trivial.

Try working it out with pencil and paper and you'll soon see the pattern emerging.

```
10 REM MANDALA
20 REM NIGEL PETERS
30 MODE 1
40 GCOL 0,1
50 MOVE 500,500
60 FOR X=0 TO 500 STEP 32
70 DRAW 500,1000-X
80 DRAW 500-X,500
90 DRAW 500,X
100 DRAW 500+X,500
110 NEXT
```

10,20 REM statements

30 Choice of mode

40 Choice of colour

50 Graphics cursor instruction

60,110 FOR...NEXT loop

Position of lines to be drawn

10-20 The usual REM statements giving information about the program to humans but not to the Electron.

30 Puts the Electron into Mode 1, allowing fine lines to be drawn. Try out other modes.

40 GCOL 0,1 chooses red as the colour the lines will be drawn in. Try using 2 or 3 instead of 1. Why don't you use 0, the other logical colour available in this mode?

50 Moves the graphics cursor to point 500,500. The drawing starts here.

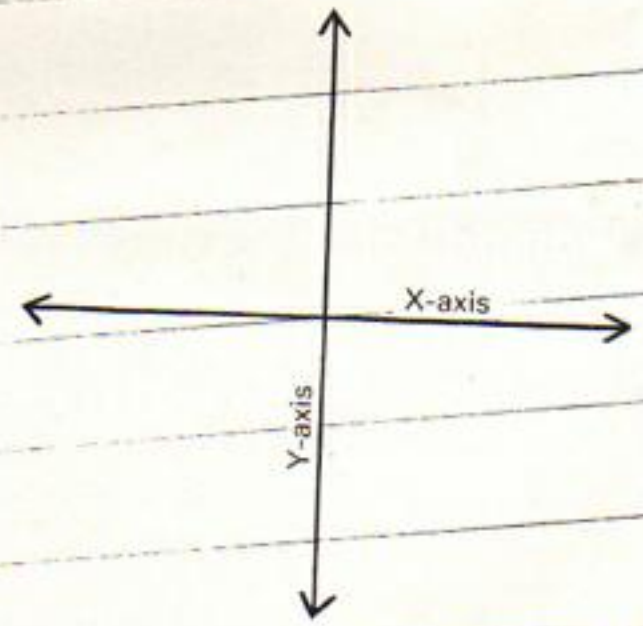
60-110 These lines define a FOR... NEXT loop which draws the pattern. Each time round the loop, four lines are drawn, the changing values of X changing the positions of the lines.

70 Draws a line to the point defined. Each time round the loop the point will move down from the top of the screen along the Y-axis.

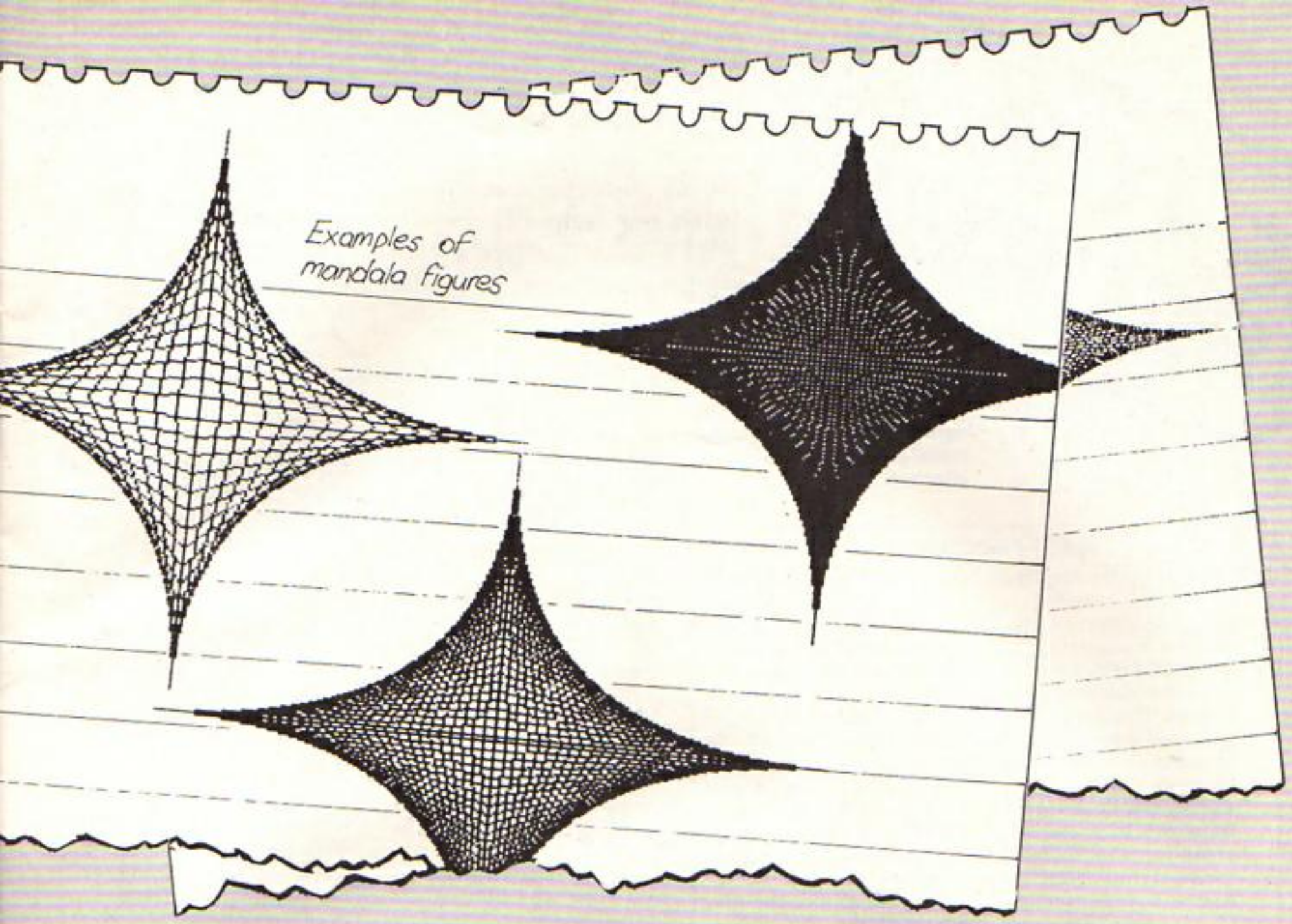
80 Draws lines from the last point to a point that moves out to the left along the X-axis each time round.

90 Draws lines to a point moving upwards each time along the Y-axis.

100 Draws lines to a point moving out along the X-axis each time round the loop.



Trevor Roberts



TOP SECRET

Send secret messages with the help of PETE DAVIDSON's coding program

THIS program is intended to be used by secret agents to send messages to other agents. Or, alternatively, from one Electron user to another.

MAIN PROCEDURES

PROCINIT: Reads the password. Change it in the last line of the program if you wish.

PROCIDENTIFY: Lets you enter the password, and checks it against the password in memory. If it is wrong three times, the program falls into an endless loop at line 350. Once you have the program working you can make it more secure against unauthorised use by inserting two more lines:

```
5 *FX200,3
6 ON ERROR GOTO 350
```

Line 5 causes memory to be wiped when Break is pressed (so that no one can list your password). Line 6 will put the program in an endless loop if escape is pressed.

Note that you must never put lines like this into any program unless it is saved on tape, and you are sure that it is error free.

PROCNUMBER: This takes in your code number and checks that it is valid. If it is you proceed. If it's not, PROCINVALID is called.

PROCINVALID: Prints out how you can create a valid code number if you type in an invalid one. It also gives you some examples to use if you cannot create your own.

PROCTYPEIN: The input procedure. The message (whether coded or not) is returned as MESSAGE$.

PROCCODE: This allows you to type your message in (using PROCTYPEIN). It then converts it first to CODE$ (by rearranging the blocks of five letters) and then to FINALCODE$ (by adding a number to the Ascii code of the letters). The procedure then gives you the option to save your code on tape.

PROCDECODE: This reads FINALCODE$ from tape, or uses PROCTYPEIN to obtain the coded message as MESSAGE$ from the keyboard and then calls it FINALCODE$.

The messages can be sent as coded writing, or as a coded message on tape. Either way it will be difficult for anyone without the correct code number to decode it.

To prevent unauthorised use of the program you have to enter a password before you can use it.

The password does not appear on the screen, so no one can read it over your shoulder.

The password we use here is ELECTRON. But it is easily changed by altering the last line of the program.

You then enter the code number. The way the message is coded depends on this number.

Either read and understand how the number works (below) or type any number. If it is not valid the computer will give you some suitable suggestions.

Remember the number! You cannot decode your message without it.

The message is split into groups of five letters. The first five numbers of the code number are used to determine the order that the letters are placed within the group.

For example, ABCDE becomes ACDBE using 13425 as the first part of the code number.

The sixth number in the code number determines how many are added to the Ascii value of each letter.

For example, if this number is 2, any As in the code become Cs.

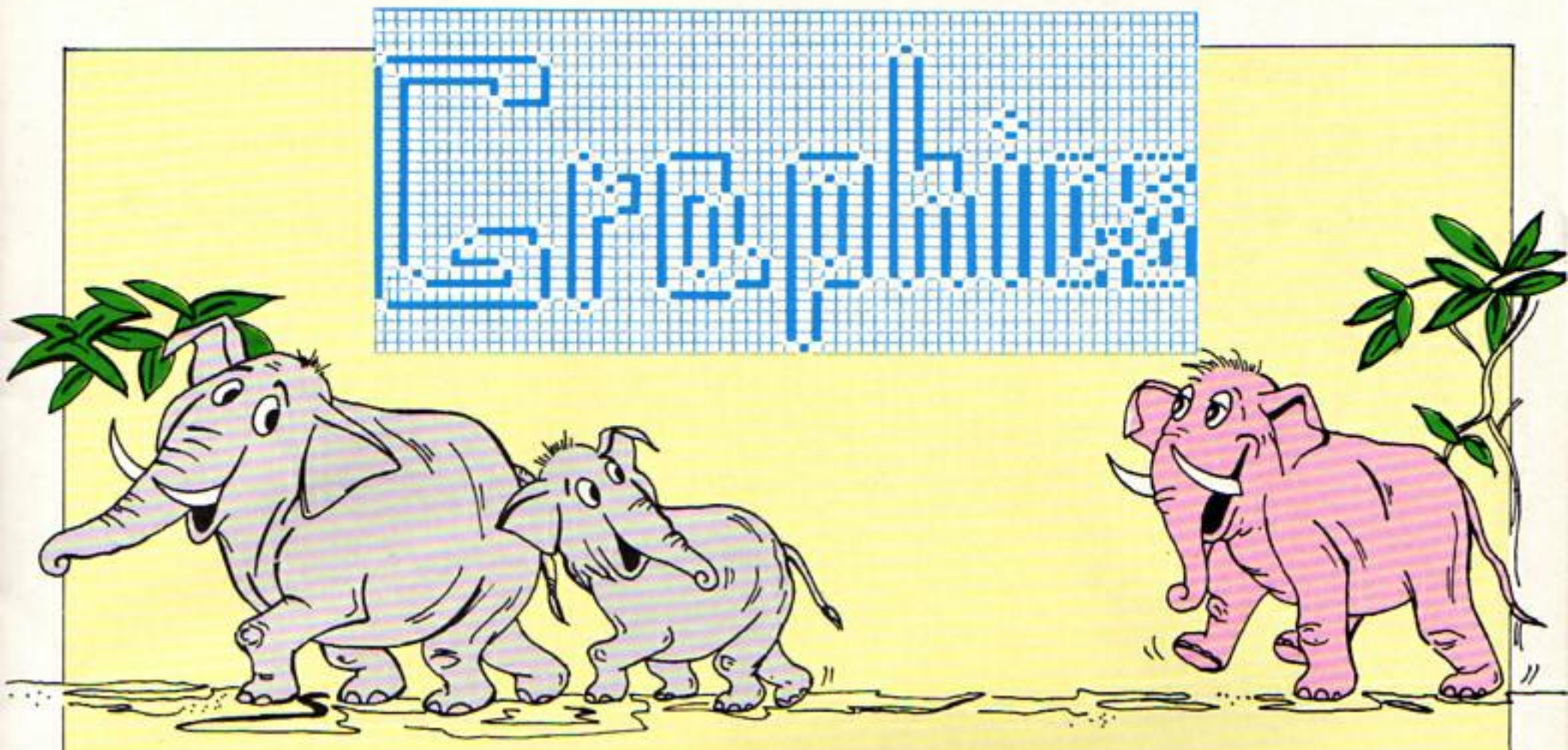
All the above means that the code number must be six numbers long, and consist of the numbers 1 to 5 in some order, followed by a number between 1 to 4. Here are some example code numbers, and a sample of how they would code ABCDEFGHIJ:

- *123450 would leave the message uncoded.*
- *123451 would change ABCDEFGHIJ to BCDEFGHIJK.*
- *543210 would change ABCDEFGHIJ to EDCBAJIHGF.*
- *543211 would change ABCDEFGHIJ to FEDCBKJIHG.*

There are 600 possible code numbers, ranging from uncoded to difficult to decode.

```
10 REM EDDIE'S CODING PROGRAM
20 REM (c) ELECTRON USER
30 REM BY PETE DAVIDSON
40 MODE 6
   :VDU 23;8202;0;0;0;
50 PROCINIT
60 PROCIDENTIFY
70 PROCNUMBER
80 CLS
   :VDU 7
   :INPUT '''"Do you want
    to code or decode? "TASK
   $
90 IF TASK$="CODE" OR TASK$=
   "code"PROCCODE
   ELSE IF TASK$="DECODE"
    OR TASK$="decode"
   PROCDECODE
   ELSE GOTO 80
100 VDU 7
   :PRINT '"Do you want
    to use the progran again
    ?"
   :ANSWER$=GET$
110 IF ANSWER$="N" CLS
   :END
   ELSE IF ANSWER$<>"Y"
   THEN 100
120 VDU 7
   :CLS
   :PRINT '"Do you want
    to use the same code
    number?"
   :ANSWER$=GET$
130 IF ANSWER$="Y"
   THEN 80
   ELSE IF ANSWER$="N"
   THEN 70
   ELSE 130
140 END
150
```

Turn to Page 60



You don't need to stick with just black and white in the two-colour modes. MIKE McMANUS encourages you to . . .

Change your colour codes!

LAST month we looked at how we could get more colour on the TV screen. We looked at the COLOUR command in particular and saw how it could be used to produce multicoloured text in Modes 1, 2 and 4.

We found that in these modes we didn't have to be stuck with the dreary old black and white default colours but could use code numbers after the COLOUR command to pick other text colours.

However we didn't do anything about the two-colour modes – 0, 3, 4 and 6.

We know from our experience with Mode 2 that we can get 16 colours on the screen, eight of them flashing.

Is it possible to have some of the more interesting colours such as, say, blue and yellow, rather than the black and white of a two-colour mode?

The answer is yes. You can choose different colours for the two-colour modes.

To do it you use the VDU19 command. This tells the Electron's operating system that you want to change the colours that are appearing on the screen.

MODES 0, 3, 4, 6

Logical number		Colour (on entering mode)
Fore-ground	Back-ground	
0	128	Black
1	129	White

MODES 1, 5

Logical number		Colour (on entering mode)
Fore-ground	Back-ground	
0	128	Black
1	129	Red
2	130	Yellow
3	131	White

MODE 2 (and actual colours)

Logical number		Colour (on entering mode)
Fore-ground	Back-ground	
0	128	Black
1	129	Red
2	130	Green
3	131	Yellow
4	132	Blue
5	133	Magenta
6	134	Cyan
7	135	White
8	136	Flashing black-white
9	137	Flashing red-cyan
10	138	Flashing green-magenta
11	139	Flashing yellow-blue
12	140	Flashing blue-yellow
13	141	Flashing magenta-green
14	142	Flashing cyan-red
15	143	Flashing white-black

The logical colour numbers on entering mode 2 are also the actual colour numbers.

Figure 1

Before we go into that, though, let's just have a look at the colours and colour codes in Mode 6.

Anything we say about this mode will apply equally as well to the other two-colour modes we've mentioned.

We know that Mode 6 is a two-colour mode. When we enter it we have two colours, white letters on a black background.

A glance at Figure 1 – which should be familiar from the last article – shows that the colour code number, or more formally, the logical colour number, is 0 for black and 1 for white.

If we were daft enough we could use these colour code numbers to give us black text on a black background.

Entering:

```
COLOUR 0
```

and pressing the Return key will have this effect. We could now get a white background by entering:

```
COLOUR 129
```

if only we could see what we're doing.

From all that you should see that the COLOUR command, coupled with the appropriate code number, allows us to mess about with the screen.

However as we only have two colour codes available in the two-colour modes, the scope isn't as great as in the other modes. We're stuck with 1 and 0.

But wouldn't it be nice if, instead of the 0 being the code for black, it could be the code for blue? And wouldn't it be good if the 1 that was the code for white could be made to represent, say, yellow?

Not only would it be nice, it's also very easy to do!

The point to grasp is that although you can only have two colours on the screen at any one time in a two-colour mode, they can be any of the 16 colours that the Electron can produce.

We came across the 16 – eight steady colours and eight flashing ones – last month.

Well, you can't have all 16 on the screen at once in Mode

→

Make light work of listings!

To save your fingers most of the listings in *Electron User* have been put on tape. Five are now available – for the February, March, April and May issues, plus a bumper tape of all the programs from the first four introductory issues.

On the May tape:

RALLY DRIVER High speed car control. **SPACE PODS** More aliens to annihilate. **CODER** Secret messages made simple. **FRUIT MACHINE** Spin the wheels to win. **CHASER** Avoid your opponent to survive. **TIC-TAC-TOE** Electron noughts and crosses. **ELECTRON DRAUGHTSMAN** Create and save Electron masterpieces. **SHEEP** A program for insomniacs. **MATHS HIKE** Mental arithmetic on the move. **MESSAGE** VDU commands in action. **ROTATION and STAR** Two graphics demonstrations. **MANDALA** The Notebook program. **PLUS LOTS, LOTS MORE.**

On the April tape:

SPACEHIKE A hopping arcade classic. **FRIEZE** Electron wallpaper. **PELICAN** Cross roads safely. **CHESSTIMER** Clock your moves. **ASTEROID** Space is a minefield. **LIMERICK** Automatic rhymes. **ROMAN** Numbers in the ancient way. **BUNNYBLITZ** The Easter program. **DOGDUCK** The classic logic game. **NOTEBOOK** Coloured grids. **BINARY** A base program.

On the March tape:

CHICKEN Let dangerous drivers test your nerve. **COFFEE** A tantalising word game from Down Under. **PARKY'S PERIL** Parky's lost in an invisible maze. **REACTION TIMER** How fast are you? **BRAINTEASER** A puzzling program. **COUNTER** Mental arithmetic can be fun! **PAPER, SCISSORS, STONE** Out-guess your Electron. **CHARACTER GENERATOR** Create shapes with this utility. **FUNNY POLYGONS** Fast graphics going round in circles. **RABBITS** Easter bunnies all over! **DRAW** Multi-coloured lines. **MEAN** Just an average program.

On the February tape:

NUMBER BALANCE Test your powers of mental arithmetic. **CALCULATOR** Make your Electron a calculator. **DOILIES** Multi-coloured patterns galore. **TOWERS OF HANOI** The age old puzzle. **LUNAR LANDER** Test your skill as an astronaut. **POSITRON INVADERS** A version of the old arcade favourite. **MOON RESCUE** Avoid the asteroids and save the spacemen. **STARS** A program making pretty pictures. **TAPESTRY** Symmetry and colour combine.

On the introductory tape:

ANAGRAM Sort out the jumbled letters. **DOODLE** Multicoloured graphics. **EUROMAP** Test your geography. **KALEIDOSCOPE** Electron graphics run riot. **CAPITALS** New upper case letters. **ROCKET, WHEEL, CANDLE** Three fireworks programs. **BOMBER** Drop the bombs before you crash. **DUCK** Simple animation. **METEORS** Collisions in space. **COMBINATIONS** Crack the hidden code. **BUZZ WORD GENERATOR** Let the Electron help you impress. **SIMON** Reactions and memory put to the test. **3-D PLOT** Enter a new dimension. **PLUS LOTS MORE!**

HOW TO ORDER

Please send me the following *Electron User* cassette tapes:

Twelve programs from the May issue ..... £

Eleven programs from the April issue ..... £

Twelve programs from the March issue ..... £

Nine programs from the February issue ..... £

26 programs from the introductory issues ..... £

I enclose the sum of £

Name .....

Address .....

.....

POST TO: Tape Offer, Electron User, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

Graphics

From Page 23

6, but you can have any two of them.

All you do is tell the micro that the colour code number 0 will in future mean green or red or whatever, while the colour code number 1 will in future stand for blue or some contrasting colour.

This is done with the VDU 19 command mentioned earlier.

Let's try it in action. Put your Electron into Mode 6 – or 0, 3, or 4 if you want. Now see if we can swap from the boring old white letters on a black background to yellow letters on a blue background.

Type in the following:

```
VDU 19,0,4,0,0,0
```

and press Return. If you've done it correctly you should see all the parts of the screen that were black turn to blue.

Now enter:

```
VDU 19,1,3,0,0,0
```

and you should see the parts that were white turn to yellow.

It's very important that you type these VDU commands in accurately as a slight error in the typing can cause chaos on the screen.

Now try typing in something on the Electron and you'll see that the foreground colour is now yellow while the background colour is blue.

What's happened is that the first VDU 19 we typed in told the micro that in future the colour that corresponded to code 0 would now be blue.

Magically anything that had been put on the screen in the colour coded 0 when it was black now turns to blue.

The second VDU 19 told the Electron that from now until further notice the colour associated with the colour code 1 would be yellow.

Again, all the previously printed white parts of the screen magically turn to yellow.

If you think about it, this has to be the case.

Mode 6 is a two colour mode, so as soon as we pick new colours for the foreground and background the old colours have to change. If they didn't there'd be more than two on the screen at any one time.

The format of the VDU 19 statement is very simple. It's just:

```
VDU 19,code number,
palette number,0,0,0
```

Or, rather more formally:

```
VDU 19, logical colour
number, actual colour
number,0,0,0
```

The VDU 19 part tells the Electron that you want to change the colours that are attached to the colour codes.

The next number is the code number of the colour that you want to change. In Mode 6 this will be either 0 or 1.

The palette number, or actual colour number, is the number that identifies the colour we will actually be using.

It would be nice if we could just tell the Electron:

```
VDU 19 Black,Blue,0,0,0
```

and then have all the background turn to blue. Sadly we can't do it like that. We have to use numbers.

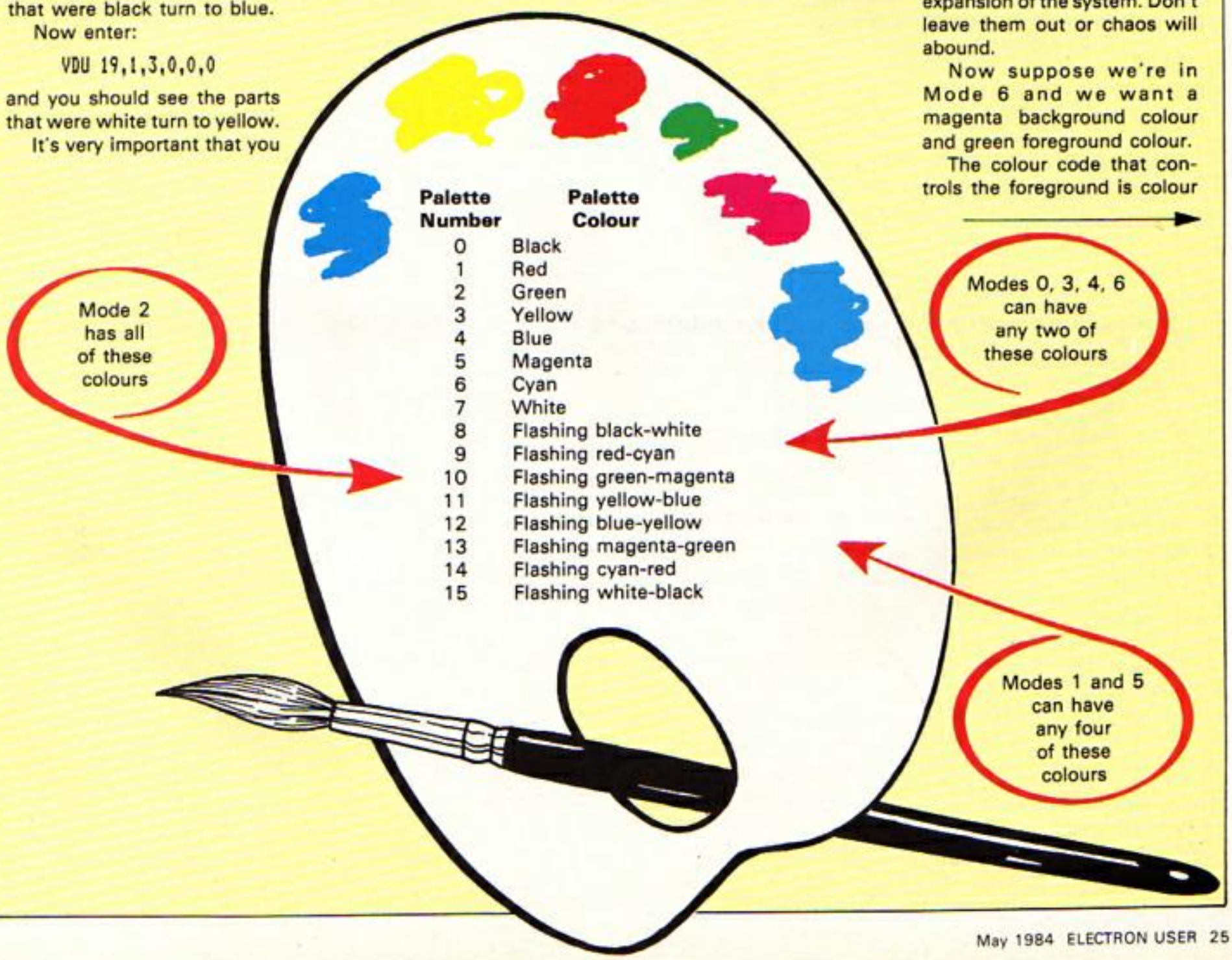
The first number is easy. It's just the colour code number that we've used along with the COLOUR command.

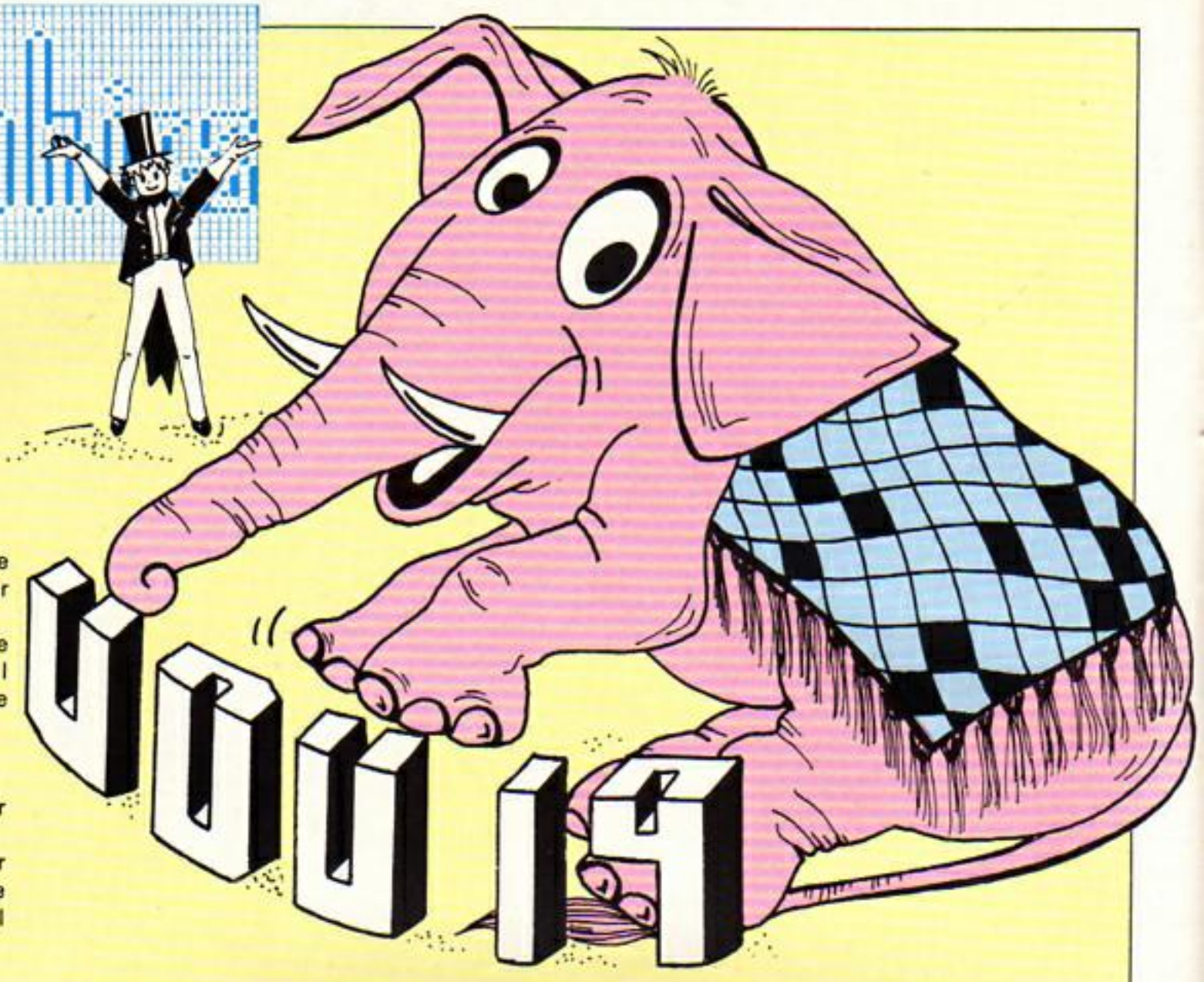
The second number, the palette number, is the number that identifies which of the 16 available colours we want. These numbers are listed in Figure II.

The three final zeroes have to be there, allegedly for future expansion of the system. Don't leave them out or chaos will abound.

Now suppose we're in Mode 6 and we want a magenta background colour and green foreground colour.

The colour code that controls the foreground is colour





From Page 25

code 1. (Figure I shows the colour codes or logical colour numbers.)

I want the foreground to be green so I look at Figure II which tells me that the palette number for green is 2. So:

```
VDU 19,1,2,0,0,0
```

will make colour code 1 refer to green.

Now the background colour code is 0 and the palette number of magenta is 5 so I want:

```
VDU 19,0,5,0,0,0
```

Horrible isn't it? If you get tired of the way you've set up the screen and want to get back to the default colours all you have to do is enter:

```
VDU 20
```

This sets the colours back to normal.

So to recap, in the two colour modes we can only have two colours on screen at any one time. However we are not stuck with the normal default colours of black and white.

We can pick any of the 16 colours that are available in Mode 2 but we can only have two of them.

We select the new colours that we want by using the VDU command. This assigns new colours to the colour codes allowed for that mode.

Program I illustrates this assignment of colours using VDU 19.

The FOR . . . NEXT loop in lines 40 to 90 changes the foreground colour – code 1 – to each of the 15 available colours in turn.

The loop in lines 110 to 160 does the same for the background colour, code 0.

Of course what applies to the two colour modes applies to Modes 1 and 5, the four colour modes. The difference is that with these modes you have four colour codes – 0, 1, 2 and 3 – to play with.

Normally these are black, red, yellow and white, but you can alter them to more exotic colours using the VDU 19 command in exactly the same way as before.

Hence if we're in Mode 5 and we want the colour coded 1 to be blue instead of the usual red we enter:

```
VDU 19,1,4,0,0,0
```

and all the red turns to blue.

Of course in Mode 2 we've already got our allocation of 16 colours so the code numbers (0 to 15) are exactly the same as the palette numbers.

Now before you read on just try all this out on your Electron.

```
10 REM PROGRAM I
20 MODE 6
30 VDU 23,1,0;0;0;0;
40 FOR palette=0 TO 15
50 VDU 19,1,palette,0,0,0
60 PRINT TAB(5,10)"code
    number 1"
70 PRINT TAB(5,13)"palette
    number ";palette
80 FOR delay=1 TO 2000
   :NEXT
90 NEXT
100 VDU 20
    :CLS
110 FOR palette=0 TO 15
120 VDU 19,0,palette,0,0
    ,0
130 PRINT TAB(5,10)"code
    number 0"
140 PRINT TAB(5,13)"palette
    number ";palette
150 FOR delay=1 TO 2000
    :NEXT
160 NEXT
```

Play around with the colours for a while, using the COLOUR command we covered last month and experimenting with the VDU 19 command.

It only takes a little practical experience to get the hang of changing colours. A concept that can appear difficult on paper soon becomes easy when you try it out for yourself.

Remember that each mode only allows a limited number of colours on the screen. The Electron isn't bothered which of the 16 colours it can produce are used in any mode. But you can only have that mode's ration.

This means that you can only have two colours on screen in Modes 0, 3, 4 and 6, four colours in Modes 1 and 5 and only in Mode 2 are you allowed the full allocation of 16.

To sum up, each mode has its ration of colour code numbers. These are the numbers we used last month after the COLOUR command.

When we enter a mode these colour codes are assigned to the default colours of that mode. We can, however, reassign them to any of the 16 colours using the VDU 19 command.

We don't have to be stuck with colour 2 being yellow in Mode 5. We can make colour 2 cyan with:

```
VDU 19,2,6,0,0,0
```

and from now on the command:

```
COLOUR 2
```

will produce cyan text – and any previous text in colour 2 will turn from red to cyan.

Have fun experimenting. If things get confused remember you can undo your VDU 19s with a VDU 20.

That's all for this month. In my next article I'm going to explore some of the uses of VDU 19.

In the meantime, why not think about this: *Why should I use a VDU 19 to assign all my Mode 2 colour codes (0 to 15) to be black (palette colour 0)?*

I'll tell you next time.

CHASER!!

EVER had one of those days when you seem to have done nothing but chase your tail?

Well now you have a chance to chase someone else's tail in this two player game for your Electron.

All the instructions are in the listing. You'll find it easy to learn and fun to play.

So type it in and get chasing.

By PETER MITCHELL and JAMES McPHERSON

30	Defines sound envelope 1.
50	Gets rid of flashing cursor.
60	Defines foreground and background colours for introduction.
80	Calls procedure intro.
90-120	Selects mode and gets rid of flashing cursor. Selects a colour.
130-160	Defines arrow characters.
170-210	Defines variables.
220	Joins text and graphics cursor.
260-330	Checks keys to see if they are being pressed.
340-370	Alters main variables.
380-450	Prints arrows and checks if there is an obstruction in their way.
460-500	Makes background sound increase in pitch as time goes on.
540-620	When called this procedure will play the notes in W$ in order and each note will have a length of L%. For example typing PROCs ("ABCD",4) will make the computer play an A, B, C, and D in order.
650-700	Checks if player one cannot move.
730-820	If player two's arrow arrives at an obstruction, tells computer which way arrow can go.
850-920	Checks if player two cannot move.
930-1020	If player one's arrow arrives at an obstruction, tells computer which way arrow can go.
1050-1090	Checks to see if it is a draw.
1130-1140	Prints who has won.
1150	Defines notes for "He's a Jolly Good Fellow".
1160-1230	Adds to relevant player's score.
1260-1640	Prepares computer for starting new game.
1780-1910	Prints players' scores.

Main Variables

X1% & Y1%	Coordinates of player one's arrow.
X2% & Y2%	Coordinates of player two's arrow.
C1%	Player one's arrow character.
C2%	Player two's arrow character.
X3% & Y3%	Direction of player one's arrow.
X4% & Y4%	Direction of player two's arrow.
SO%	Pitch of background sound.
G%	Number of wins player one has had.
H%	Number of wins player two has had.

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
10 REM CHASER
20 REM (C) ELECTRON USER
30 ENVELOPE 1,1,1,-1
   ,1,10,10,10,126,0
   ,0,0,75,75
40 MODE 6
50 VDU 23;8202;0;0;0;
60 VDU 19,0,4,0;0;19
   ,1,3,0;0
70 G%=0
   :H%=0
80 PROCIntro
90 MODE 1
100 SO%=0
110 VDU 23;8202;0;0;0;
120 VDU 19,3,10,0,0,0
130 VDU 23,225,24,60,126
   ,219,153,153,153,153
140 VDU 23,226,153,153
   ,153,153,219,126,60
   ,24
150 VDU 23,227,248,12
   ,6,255,255,6,12,248
160 VDU 23,228,31,48,96
   ,255,255,96,48,31
170 X1%=RND(10)
   :Y1%=RND(31)
180 X2%=RND(10)+30
   :Y2%=RND(31)
190 C1%=225
   :C2%=226
200 X3%=0
   :X4%=0
210 Y3%=1
   :Y4%=-1
220 VDU 5
230 TIME =0
240 REPEAT
250 X11%=X1%
   :Y11%=Y1%
260 IF INKEY (-98)
   THEN X3%=-1
   :Y3%=0
   :C1%=228
270 IF INKEY (-67)
   THEN X3%=1
   :Y3%=0
   :C1%=227
280 IF INKEY (-103)
   THEN X4%=-1
   :Y4%=0
   :C2%=228
290 IF INKEY (-104)
   THEN X4%=1
   :Y4%=0
   :C2%=227
300 IF INKEY (-66)
   THEN Y3%=-1
   :X3%=0
   :C1%=226
310 IF INKEY (-17)
   THEN Y3%=1
   :X3%=0
   :C1%=225
320 IF INKEY (-105)
   THEN Y4%=-1
   :X4%=0
   :C2%=226
330 IF INKEY (-73)
   THEN Y4%=1
   :X4%=0
   :C2%=225
340 X1%=X1%+X3%
350 X2%=X2%+X4%
360 Y2%=Y2%+Y4%
370 Y1%=Y1%+Y3%
380 IF POINT(X1%*32+16
   ,Y1%*32-16)<>0
   THEN PROCc
390 GCOL 0,1
400 MOVE X1%*32,Y1%*32
410 VDU C1%
420 IF POINT(X2%*32+16
   ,Y2%*32-16)<>0
   THEN PROCc2
430 GCOL 0,2
440 MOVE X2%*32,Y2%*32
450 VDU C2%
460 UNTIL TIME >100
470 SO%=SO%+5
480 SOUND 1,1,SO%,1
490 TIME =0
500 GOTO 240
510 END
520 :
530 :
540 DEF PROCs(W$,L%)
550 Q$="A BC D EF G"
560 RE$="a bc d ef g"
570 FOR Q%=1 TO LEN (W$)
580 E$=MID$(W$,Q%,1)
590 NU%=INSTR(Q$,E$)
600 IF NU%=0
   THEN NU%=INSTR(RE$
   ,E$)
   :SOUND 1,-13,NU%+4+37
   ,L%*2
   ELSE SOUND 1,-13,NU%+4+
   37,L%
610 NEXT
620 ENDPROC
630 :
640 :
650 DEF PROCc
660 PROChelp
670 IF X3%=0 AND Y3%=0
```

Turn to page 59

beware of low-flying sheep!

REMEMBER Bright Eyes, the program from Mike Rowe we featured in the March *Electron User*?

We asked what next and wondered whether readers would send us programs with sheep leaping over gates.

Well they have – or at least, Mike Rowe has risen to the occasion and sent us one. He doesn't tell us whether or not he's an insomniac!

Now, who's going to send us a low flying pig?

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
10 REM Sheep jumping
   over a fence
20 REM From same hole
   as 'RABBITS'
30 REM By Michael Rowe

40 REM (C) ELECTRON USER
50 MODE 2
60 VDU 23,0,8202;0;0;0;
70 VDU 19,128,132,0,0
   ,0
   : REM Sky
80 GCOL 0,2
   : REM field
90
100 MOVE 0,0
   : MOVE 0,500
   :PLOT 85,1280,500

110 MOVE 0,0
   : MOVE 1280,500
   :PLOT 85,1280,0
120
130 PROCchrs
140
150 FOR Y=10 TO 15
160 PRINT TAB(7,Y) "Z"
   : REM wall
170 NEXT Y
180
190 PROCsheep(16,13)
200 PROCsheep(13,13)
210 PROCsheep(10,10)
220 PROCsheep(7,7)
230 PROCsheep(4,10)
240 PROCsheep(1,13)
250 PROCsheep(0,13)
260 GOTO 190
270 END
280 **********
290 DEF PROCchrs
300 VDU 23,224,4,63,111
   ,255,126,30,15,7
310 REM Mid top
320 VDU 23,225,0,0,0,0
   ,0,0,181,255
330 REM Rump
340 VDU 23,226,0,0,0,0
   ,0,0,82,244
350 REM Front
360 VDU 23,227,7,15,7
   ,31,15,31,15,7
370 REM Bot mid
380 VDU 23,228,255,255
   ,255,255,255,255,255
   ,26
390 REM Bot back
400 VDU 23,229,245,248
   ,248,248,240,240,240
   ,240
410 VDU 23,229,245,248
   ,248,248,240,240,240
   ,240
420 REM Front legs 1st
430 VDU 23,230,2,2,2,3
   ,0,0,0,0
440 REM Front legs 2nd
450 VDU 23,231,0,0,0,192
   ,64,0,0,0
460 REM Back legs
470 VDU 23,232,16,16,144
   ,240,0,0,0,0
480 ENDPROC
490
500 DEF PROCsheep(x,y)
510 PRINT TAB(x,y)
   CHR$ 224;CHR$ 225;
   CHR$ 226
520 PRINT TAB(x,y+1)
   CHR$ 227;CHR$ 228;
   CHR$ 229
530 PRINT TAB(x,y+2)
   CHR$ 230;CHR$ 231;
   CHR$ 232
540
550 REM holds sheep for
    a while
560 TIME =0
   : REPEAT UNTIL
   TIME = 35
570
580 REM rubs out sheep
590 PRINT TAB(x,y)SPC (3)
600 PRINT TAB(x,y+1)
   SPC (3)
610 PRINT TAB(x,y+2)
   SPC (3)
620 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 47

MIKE COOK illustrates rubber-banding techniques you can try on your Electron

Quick on the draw

THE Electron's superior graphics are capable of producing a very good picture. However this usually requires that the picture be expressed in mathematical form or be reduced to a series of coordinates which the computer plots.

The program given here offers an alternative approach by allowing the computer to interact with the user.

You can position lines precisely, anywhere on the screen, by using rubber-banding techniques.

This is where a line is drawn on the screen by fixing one end and then moving the other until it is in the desired position.

The line appears to grow from the fixed point like a rubber band, hence the name.

The cursor keys control a single point – or pixel – on the screen. The longer a key is held down the faster the point moves.

This permits slow and careful positioning while allowing rapid movement to another area of the screen.

Like all graphics programs on the Electron, we must decide which mode to use. This is normally a compromise between screen resolution and the number of colours available.

This program uses Mode 1 and thus allows reasonable resolution with a choice of four colours, including the background.

The program can be modified easily to use another mode if this compromise is not to your liking.

The program will run in a series of modes. In this case "mode" refers not to a graphic mode, but rather to what shape is currently being generated.

When running the program the screen will go blank. Nothing will happen until one of the "modes" is entered. To do this type the letter for the appropriate one.

For example, let's draw a straight line by pressing the L key.

The top line of the display should now say LINE, and you should see a single lit point at the bottom left hand corner of the screen.

This can be moved by using any of the four cursor keys.

When it is in the correct position for the start of the line, press any other key to fix the start of the line.

When the point is moved again a line will be drawn from the start of the line to the new position.

This line will follow the point, behaving like a rubber band on the screen.

When the end position of the line is at the correct place press the Return key and the line will be drawn in permanently.

You will then be back in the Command mode, ready to draw another shape.

If you want to carry on drawing lines press the space bar instead of the Return key and you will stay in the Line mode.

If you want the start of the new line to be the same point as the end of the old one, press "J" for join instead of the space bar.

Note that it is possible to move the point off the screen. Indeed you may want to do this intentionally.

When this occurs the co-ordinates of the point you are moving will appear on the top line of the display. This lets you know which way to go when you want to return the point to the screen.

The Triangle mode – key T – works in a similar manner, with first the base line being rubber-banded and then the full triangle.

In this mode the J key will join up the new triangle to the last side of the old one.

When drawing a rectangle – key R – the first point fixes one corner and the second point will fix the opposite corner.

The Join key will join the next rectangle to the last corner of the previous one.

When selecting the polygon mode – key P – you will be asked how many sides the polygon is to have.

If you require a circle then a large number like 40 should be used.

In this mode the radius is defined by rubber banding, but this radius line disappears when the polygon is drawn.

If the Join option is used the new polygon will be drawn with the same centre as the last one.

You can also change the colour of the lines by pressing the C key. This will cycle through the three colours available in Mode 1.

The colours have been redefined from the default choice in line 170. They could be changed to suit your own preferences.

When you have finished your masterpiece the screen can be saved as a file by typing S.

You will be asked to provide a file name and the memory locations that make up the screen will be saved. You must then put a tape in the recorder for the file.

The file is saved as a block of memory and, as this is 20k long, it takes some time to save.

The program also lets you load a previously dumped file back to the screen to be worked on further. This is done by typing L.

It can also be done from your own programs by performing a CLS command and then a *LOAD"FILENAME", using, of course, the file name employed to save the screen.

You will see the picture appear block by block on the screen.

Note that in order for this to work you have to be in the same graphics mode as the computer was when the picture was created.

However you may have different colours as defined by the VDU 19 commands.

Listing starts on Page 57

Here are some hints for typing in the program:

Line 160 defines a text window of one line at the top of the screen. This line is best left out until all the typing errors are corrected, as any error messages will scroll off the top before you can read them.

As the cursor keys are used to move the point they are not in the correct mode for editing a mistyped line.

Function key 0 has been set up – in line 40 – to restore the editing function and the auto repeat of the keys.

It should be pressed to regain these functions.

Some variables start with the letter O, such as OX% (Old X). Do not confuse this with the number 0.

In line 400 the space between the quote marks and the number is vital. You will get an error message if it is left out. Unfortunately the error message is not all that helpful.

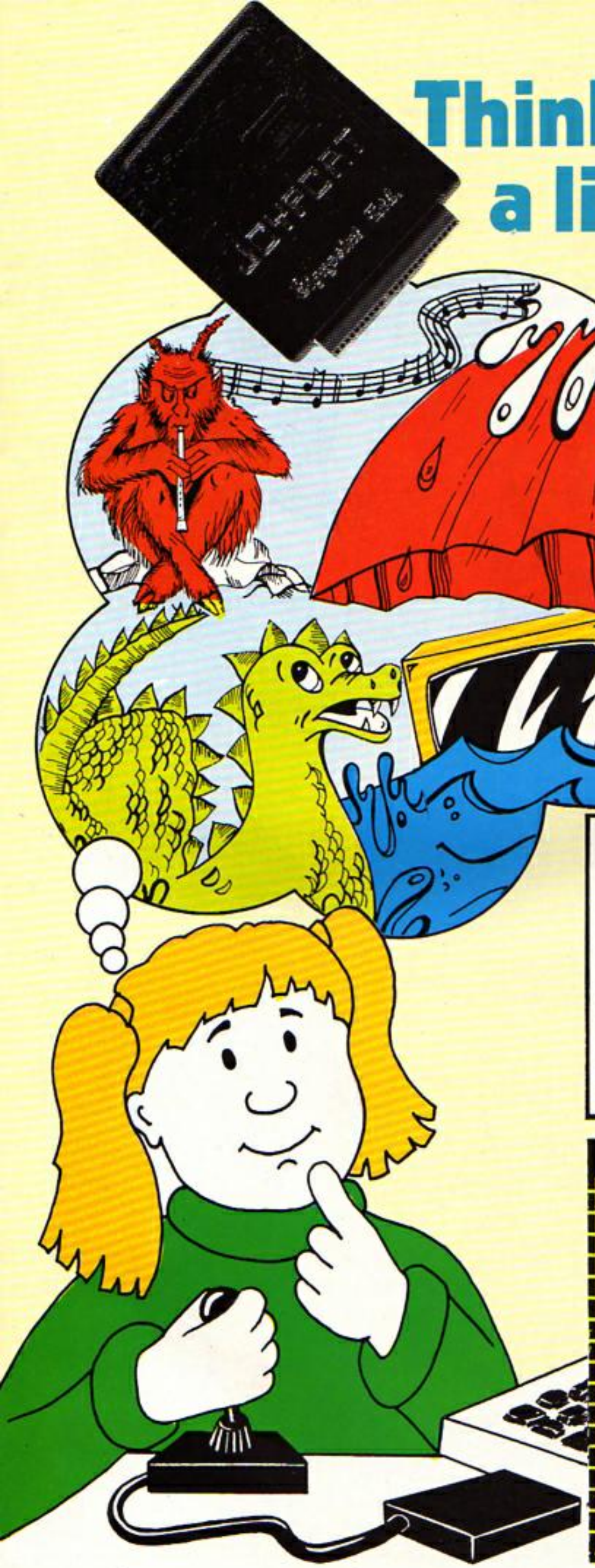
In various lines, such as line 240, note that there is no space between the quote marks. If a space is placed there the loop will end prematurely and that section will appear to do nothing.

ELECTRON USER

FREE CONTEST

Think up a link up

and WIN the Signpoint joyport!



EXERCISE your imagination and win yourself a Signpoint joyport, the interface that allows you to use joysticks with your Electron.

This month's Casting Agency contains six characters – a dragon, a devil, two musical notes, an umbrella and a TV.

Apart from the fact that they've all been sent in by our readers, and they're in May's Casting Agency, what else could possibly link them all?

Let's have your ideas on why they're all together. Then finish off the sentence in the coupon in 25 to 50 words.

The best entry received before May 30, 1984, will receive the Signpoint joyport and our winner will be able to add a whole new dimension to his Electron.

WE HAVE A WINNER

THE prize for our February competition – 12 month's supply of Acornsoft Software – attracted a record number of entries.

Mind you, since it was our first competition that was no surprise!

What was surprising was the number of you who tried to put the Electron's outstanding features into the same order as our experts.

Winner of the software was Robert Waite, of Finmere, Bucks.

Not only did his list correspond with ours, we particularly like the way he finished the tie-breaker:

"I'm glad I got my Electron because it was getting rather cold camping outside W.H. Smiths".

ELECTRON USER CONTEST

The reason all the Casting Agency characters are together is:

.....

.....

.....

.....

NAME .....

ADDRESS .....

.....

Post to: Joyport, *Electron User* Contest, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

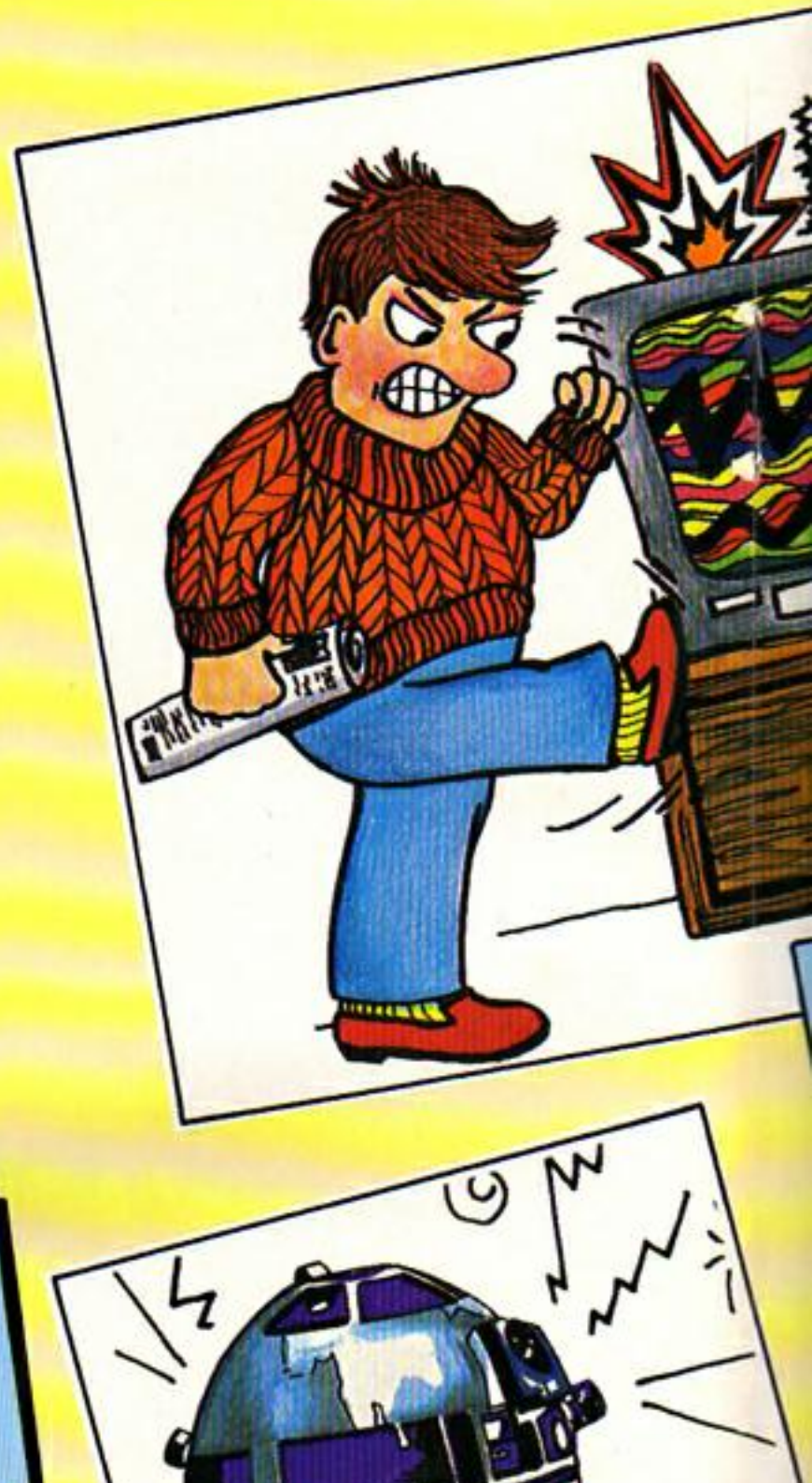
SOUNDS.. EXCITING

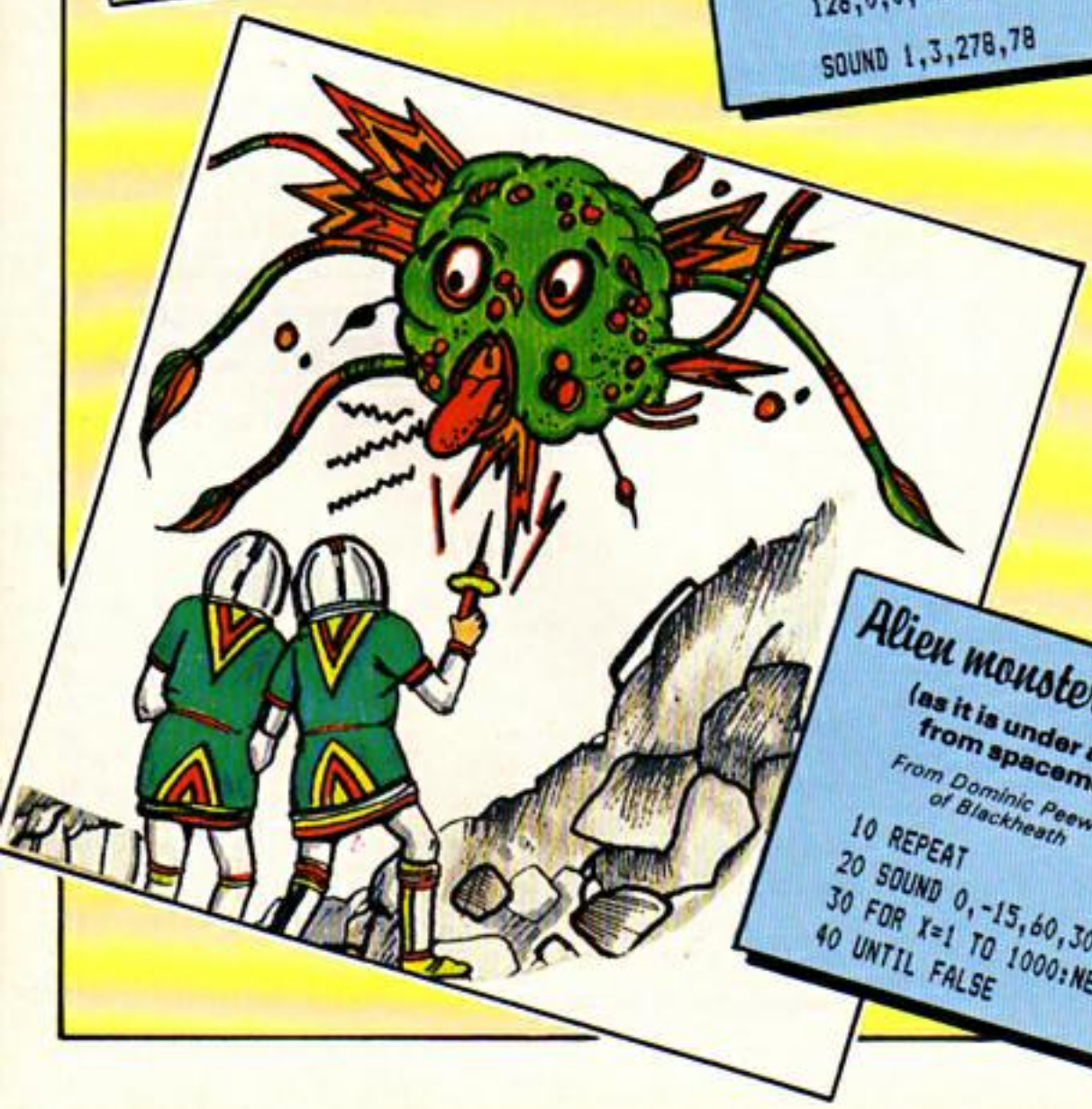
BUILD up a library of exciting sounds to enhance your own programs with these listings. And many more in the months to come!

Electric Alarm Clock

From Martin Young of Mansfield, Woodhouse

```
ENVELOPE
3,1,-36,-93,-9,11,0,0,
126,0,0,-126,126,126

SOUND 1,3,278,78
```



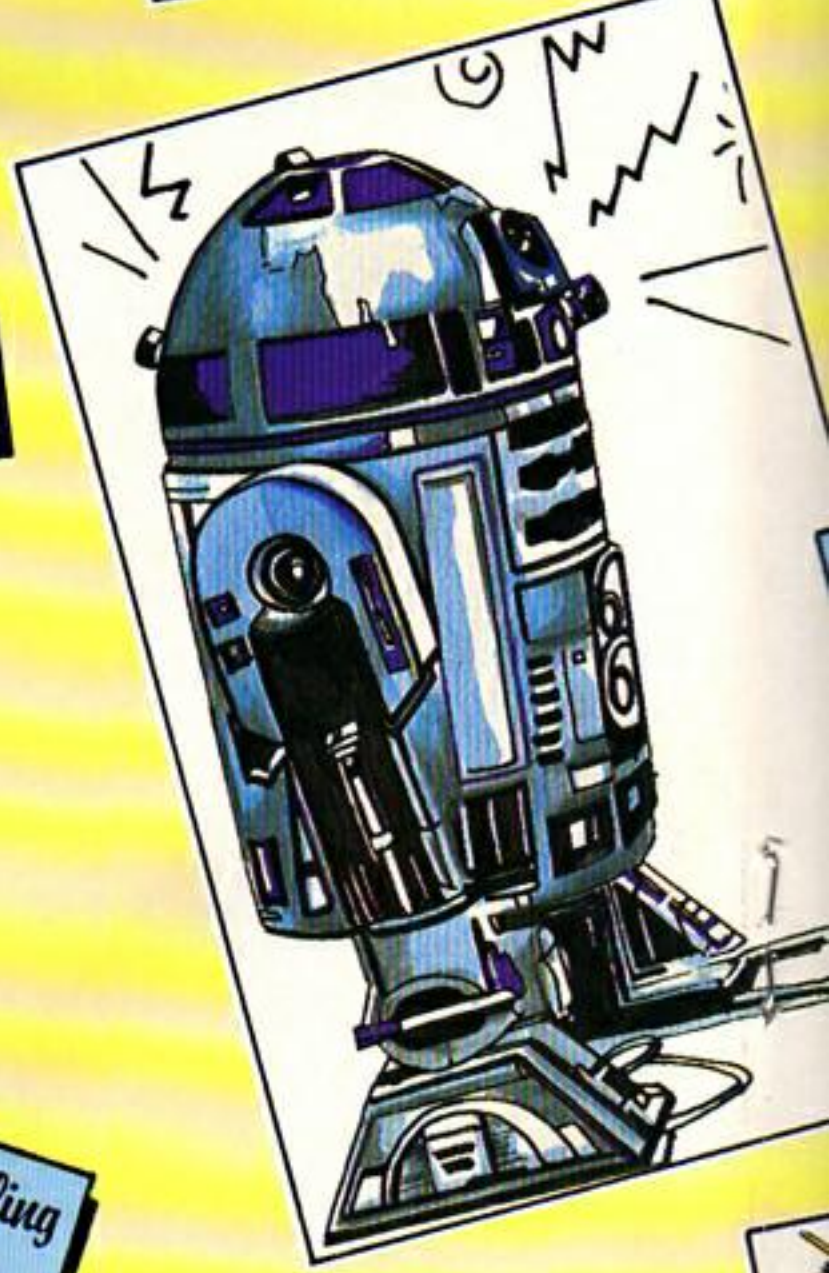


Alien monster howling

(as it is under attack from spacemen)

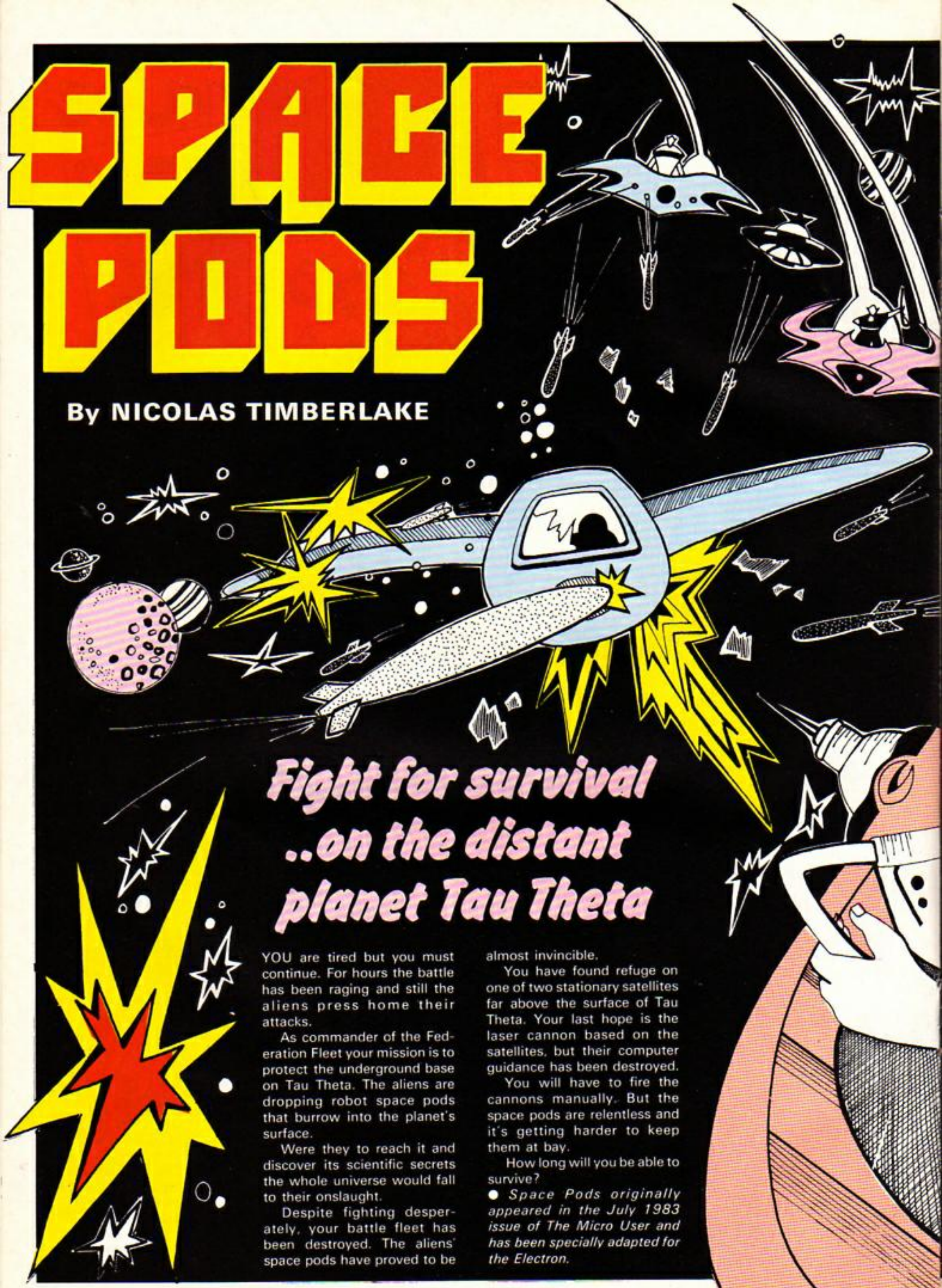
From Dominic Peewich of Blackheath

```
10 REPEAT
20 SOUND 0,-15,60,30
30 FOR X=1 TO 1000:NEXT
40 UNTIL FALSE
```



SPACE PODS

By NICOLAS TIMBERLAKE



Fight for survival ..on the distant planet Tau Theta

YOU are tired but you must continue. For hours the battle has been raging and still the aliens press home their attacks.

As commander of the Federation Fleet your mission is to protect the underground base on Tau Theta. The aliens are dropping robot space pods that burrow into the planet's surface.

Were they to reach it and discover its scientific secrets the whole universe would fall to their onslaught.

Despite fighting desperately, your battle fleet has been destroyed. The aliens' space pods have proved to be almost invincible.

You have found refuge on one of two stationary satellites far above the surface of Tau Theta. Your last hope is the laser cannon based on the satellites, but their computer guidance has been destroyed.

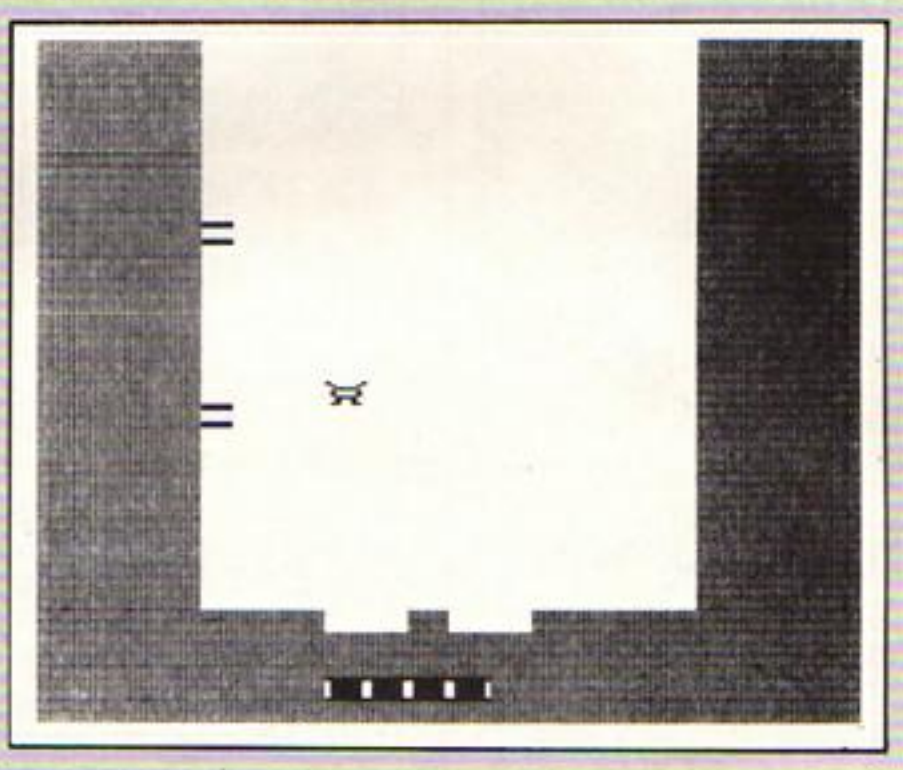
You will have to fire the cannons manually. But the space pods are relentless and it's getting harder to keep them at bay.

How long will you be able to survive?

● *Space Pods originally appeared in the July 1983 issue of The Micro User and has been specially adapted for the Electron.*

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
10 REM ****************
20 REM * SPACE PODS *
30 REM ****************
40 MODE 6
   :PRINT TAB(10,6);"SPACE
    PODS !"
50 REM (C) ELECTRON USER
60 PRINT TAB(9,9);"By N.Timb
   erlake"
70 PRINT TAB(3,16);"Do you
    want instructions(Y/N)?"
   ;
   :G$=GET$
   :IF G$="Y"
   THEN PROCINSTRUCTIONS

80 MODE 2
90 ENVELOPE 1,1,-1,0,0
   ,2,0,0,126,0,0, -10
   ,126,126
100 SCORE%=0
   :T=0
110 DEF FNpoint(X,Y)=
   POINT((64*X+32),(32*(31-Y
   )+16))
120 VDU 23,231,255,126,126
   ,126,126,126,126,255
   :B$=CHR$ 231+CHR$ 231+
   CHR$ 231+CHR$ 231
130 VDU 23,233,129,66,60
   ,66,66,60,36,102
140 VDU 23,232,252,252,0
   ,0,0,0,252,252
150 COLOUR 129
   :COLOUR 6
   :CLS
160 VDU 23;8202;0;0;0;
170 VDU 23,230,255,255,255
   ,255,255,255,255,255
180 VDU 23,240,0,0,255,0
   ,0,255,0,0
200 A%=-1
210 A%=A%+1
220 IF A%=4
   THEN A%=16
230 IF A%>19
   THEN GOTO 280
240 FOR B%=0 TO 29
250 PRINT TAB(A%,B%);
   CHR$ 230
260 NEXT B%
270 GOTO 210
280 B%=24
290 B%=B%+1
300 IF B%>29
   THEN GOTO 350
310 FOR A%=0 TO 19
320 PRINT TAB(A%,B%);
   CHR$ 230
330 NEXT A%
340 GOTO 290
350 COLOUR 0
360 PRINT TAB(7,28);B$
370 COLOUR 0
380 PRINT TAB(4,8)CHR$ 232
390 PRINT TAB(4,16)CHR$ 232
400 X%=RND(11)+4
410 Y%=-1
420 G$=INKEY$ (0)
430 IF G$="W" OR G$="X"
   THEN GOTO 530
440 Y%=Y%+1
450 IF Y%>29
   THEN PRINT TAB(X%,Y%-1)
   CHR$ 32
   :GOTO 400
460 IF FNpoint(X%,Y%)=6
   THEN PRINT TAB(X%,Y%-1)
   CHR$ 32
   :PRINT TAB(X%,Y%)
   CHR$ 32
   :GOTO 400
470 IF Y%<>0
   THEN PRINT TAB(X%,Y%-1)
   CHR$ 32
480 PRINT TAB(X%,Y%);
   CHR$ 233
490 IF Y%=28 AND X%>6
   AND X%<11
   THEN PROCEND
500 IF SCORE%>2000
   THEN T=0
510 FOR VB%=1 TO T
   :NEXT VB%
520 GOTO 420
530 REM WHICH ONE
540 SOUND 0,1,100,1
550 PROCTOP
560 GOTO 420
570 DEF PROCEXPLOSION
580 SOUND 1,1,100,1
590 SCORE%=SCORE%+250
   :PRINT TAB(2,30);"SCORE:"
   ;SCORE%
600 Y%=0
   :X%=RND(11)+4
   :ENDPROC
610 DEF PROCTOP
620 COLOUR 0
630 IF G$="W"
   THEN FY%=8
   ELSE FY%=16
640 FOR FX%=5TO 15
650 PRINT TAB(FX%,FY%)
   CHR$ 240
660 IF FX%=X% AND FY%=Y%
   THEN PROCEXPLOSION
670 NEXT FX%
680 PRINT TAB(5,FY%)"
                *
690 ENDPROC
700 DEF PROCEND
710 RESTORE
720 READ P%
730 FOR A%=1 TO 100
   :NEXT
740 IF P%=256
   THEN PRINT TAB(4,14)
   "YOU ARE DEAD"
750 IF P%=256
   THEN PRINT TAB(4,16)
   "ANOTHER GO";
   :INPUT G$
760 IF G$="Y"
   THEN GOTO 90
770 IF G$="N"
   THEN CLS
   :END
780 IF G$<>"Y" AND G$<>"N"
    AND P%=256
   THEN PRINT TAB(4,16)
   "              "
   :GOTO 750
790 IF P%=257
   THEN FOR A%=1 TO 100
   :NEXT
800 IF P%<256
   THEN SOUND 1,-15,P%
   ,1
810 GOTO 720
820 DATA 81,69,53,69,81
   ,257,69,257,61,73,49
   ,61,73,257,61,257,81
   ,69,53,69,81,257,69
   ,257,33,41,49,53,256
830 DEF PROCINSTRUCTIONS
840 CLS
850 PRINT TAB(10,3);"SPACE
    PODS !"
860 PRINT TAB(3,8);"The objec
   t of the game is to stop
   thespace pods landing
   and eating their way
   to your base.To stop
   them you have to  shoot
   them down with your
   laser guns.You have
   two laser guns which
   can be fired by pressing
   ";
870 PRINT "either'W' or 'X'.E
   very time you hit a space
    pod,you will get 260
       points."
880 PRINT TAB(3,20);CHR$ 133
   "Press any key to continu
   e";
   :G$=GET$
   :ENDPROC
```



This listing is included in this month's cassette tape offer. See order form on Page 47

LET'S GO ON A MATHS HIKE!

ONCE upon a time, many years ago when I was in primary school, our class had a maths teacher who used to take us on "Mathematical Hikes".

Of course we never left the classroom. What he meant was that he'd take us on a tour of our powers of mental arithmetic.

He would tell us the first number of the hike – suppose it was 5 – and say: "Multiply it by 2". Then he'd say something like: "Now add 7 to the total and multiply the result by 6".

When he thought he'd gone far enough he'd ask us the answer, which in this case is 102.

Occasionally he'd carry on until only one or two of us could keep the total in our heads.

Going on mathematical hikes really made mental arithmetic interesting.

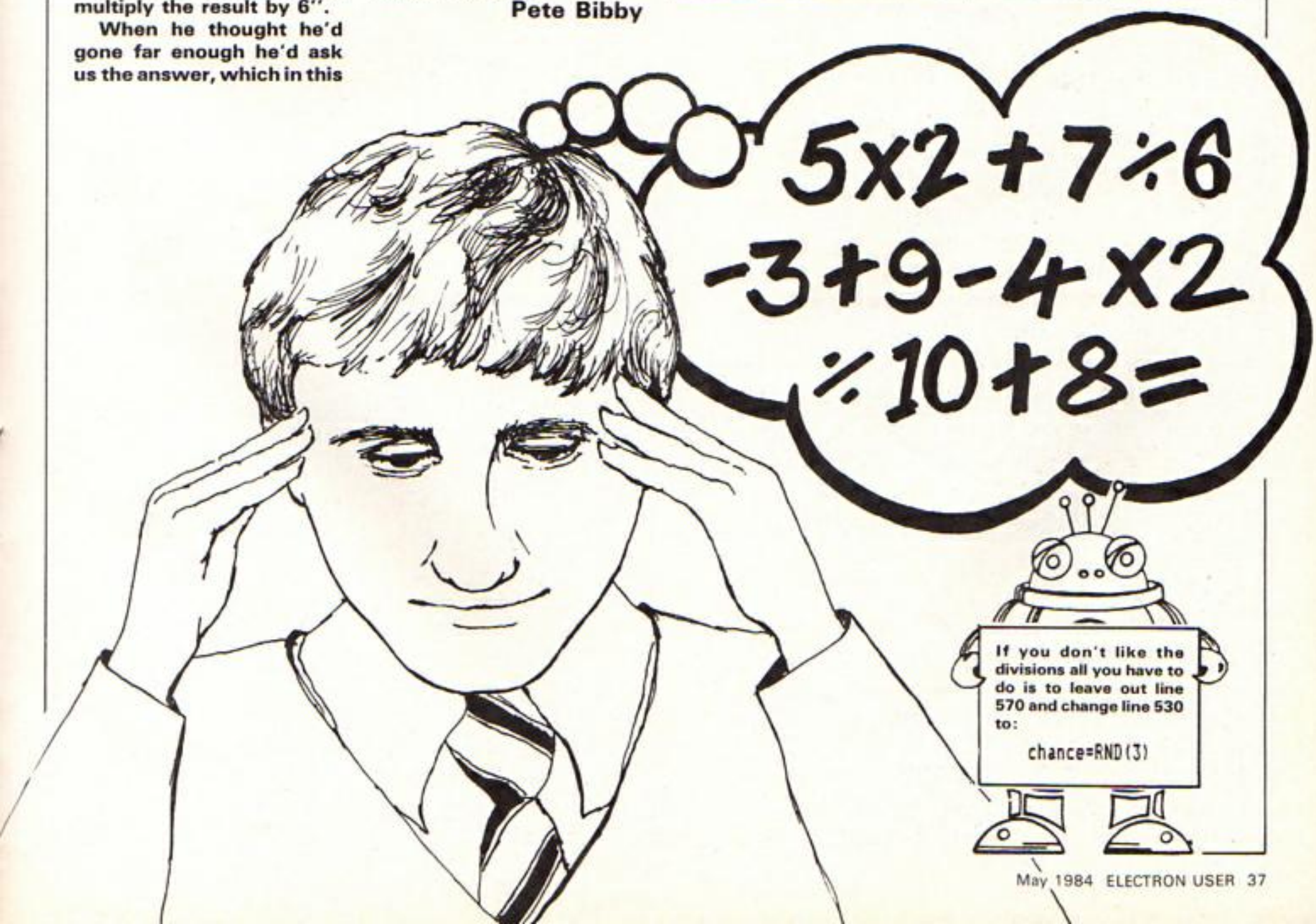
Of course you don't need a teacher to take you on mathematical hikes when you've got an Electron.

Just type in this program and your micro will play the part of the teacher. And it won't keep you in after school! Have fun.

Pete Bibby

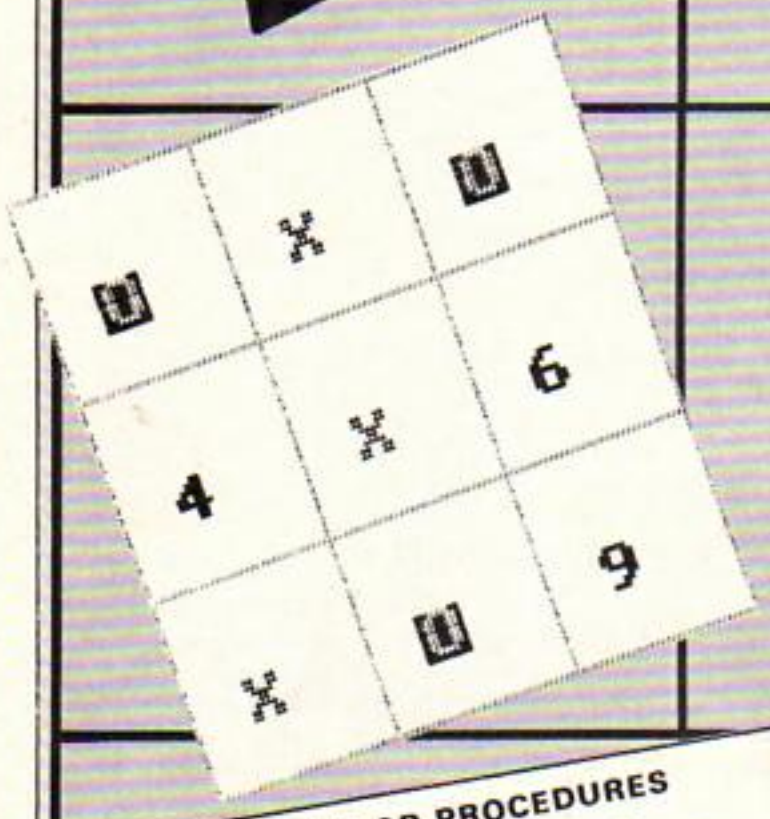
```
 10 REM MATHS HIKE
 20 REM (C) ELECTRON USER
 30 REM by Pete Bibby
 40 ON ERROR GOTO 90
 50 MODE 6
 60 VDU 19,0,4,0,0,0
 70 VDU 23,1,0;0;0;0;

 80 PROCinstruction
 90 PROCinput
100 REPEAT
110 PROChike
120 PROCanswer
130 WAIT$=GET$
140 CLS
150 UNTIL FALSE
160 END
170 DEF PROCinstruction
180 PRINT TAB(10,1) "A
    Mathematical Hike"
190 PRINT TAB(10,2) "******
    ************"
200 PRINT TAB(7,4) "Your
    Electron is now going"
210 PRINT TAB(7,6) "to
    test you on your power
    s"
220 PRINT TAB(7,8) "of
    mental arithemetic."
230 PRINT TAB(7,10) "It
    will think of a number
    "
240 PRINT TAB(7,12) "and
    then tell you to add"
250 PRINT TAB(7,14) "or
    subtract or multiply
    or"
260 PRINT TAB(7,16) "divide
    it by the numbers"
270 PRINT TAB(7,18) "that
    appear on the screen."
280 PRINT TAB(7,20) "Your
    job is to try to keep"
290 PRINT TAB(7,22) "the
    total in your head."
300 FOR delay=1 TO 2000
    :NEXT delay
    :CLS
310 ENDPROC
320 DEF PROCinput
330 CLS
340 REPEAT
350 INPUT TAB(3,5)"What
    should be the largest
    number?"TAB(20,7)limit%
360 IF limit%<=1
```

Turn to Page 58



MARK SMIDDY adapts the classic game that keeps you on your toes . . .

Tic Tac Toe



ELECTRON Tic-Tac-Toe is another version of the age old game of noughts and crosses.

There's no pencil and paper though. You just pit your wits against the Electron.

There are two skill levels. The easiest is level 1. But to beat the Electron on level 2 is almost – but not quite – impossible.

MAJOR VARIABLES

D%	Equals 0 if game ended.
L%	Equals TRUE if player lost.
N%	Used as a loop counter.
P%	Used as a loop counter.
W%	Equals TRUE if player won.
X%	X coordinate of O or X.
Y%	Y coordinate of O or X.
rnd	Equals TRUE if easy game selected.
win	Equals TRUE if computer could beat you.
block	Equals TRUE if computer could stop you.
X,Y,Z	Pointers to array elements.
x,y,z	Pointers to array elements.
Z$(n)	Array holding all the pieces on the board.

MAJOR PROCEDURES

PROCins	Prints up instructions.
PROCset	Initialises start of each game.
PROCcomputer	Decides computer's move.
PROCrnd	Makes random move for computer.
PROCplayer	Gets and checks player's move.
PROCx	Puts an 'O' into the Z$(N) array, at the position pointed to by X, Y or Z.
PROCboard	Draws the board.
PROCpieces	Draws the Os or Xs.
PROCinit	Sets up the computer for the start of play. Defines the envelopes, and picks the initial colours.

MAJOR FUNCTIONS

FNtry(1$)	Returns TRUE if the string 1$ is found using FNt
FNt	Returns TRUE if three Os or three Xs are found in a row.
FNtest	Returns TRUE if a free space is found in a line X, Y, Z.

Tic-Tac-Toe listing

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
10 REM TIC-TAC-TOE.
20 REM by Mark Smiddy
30 REM (C) ELECTRON USER
40 MODE 1
50 VDU 23,0,8202;0;0;0;
60 PROCinit
70 VDU 19,1,2;0;
80 PROCins
90 REPEAT
100 CLS
110 PRINT ''"Choose your
    skill level (1-2) "
120 REPEAT
130 rnd=GET -50
140 UNTIL rnd=0 OR rnd=-1
150 PROCset
160 CLS
170 PROCboard
180 PROCpieces
190 REPEAT
200 D%=0
210 PRINT
220 PROCplayer
230 IF (L%=0 AND W%=0)
    PROCpieces
240 IF FNtry("X")COLOUR 3
    :PRINT TAB(0,18)"You
     win"
    :W%=1
    :SOUND &11,1,20,20
250 FOR N%=1TO 9
260 IF Z$(N%)="." D%=D%+1
270 NEXT
280 IF D%=0 AND W%=0
    COLOUR 3
    :PRINT TAB(0,18)"IT'S
```

Turn to Page 53

You'll be ITCHING to get your hands on the funniest program of 1984!

Fleas!
Bed Bugs!
And all you wanted was a quiet night . . .

The pests are after your feet and you'll have to move fast to stop them. Swot them with a jam sandwich or crunch them with your false teeth.

If you're desperate you can always phone for help. But whatever you do, do it quickly. You need cunning tactics and nimble fingers!

Bed Bugs guarantees hours of hilarity for the whole family.

OPTIMA SOFTWARE

Obtainable from your Acorn dealer or send in the coupon below

ORDER FORM

Please send me BED BUGS:

- ☐ *BBC 'B' cassette – £6.95*
- ☐ *Electron cassette – £6.95*
- ☐ *BBC 40-track disc – £8.95*
- ☐ *BBC 80-track disc – £8.95*

Add 50p p&p (post free 2 or more)

Name .....

Address .....

.....

.....

.....

☐ I enclose cheque made payable to Optima Software Ltd.

☐ I wish to pay by ☐ Access ☐ Visa

No .....

Signed .....

Expiry date .....

Optima Software Ltd., 36 St. Petersgate, Stockport SK1 1HL.

Casting Agency

Fred the Dragon

From Christopher Bingham (Bristol)

```
VDU 23,230,241,223,255,
224,254,224,224,224

VDU 23,231,224,224,224,
224,160,160,160,240

VDU 23,232,1,1,3,7,
15,30,56,240
```

Television

From Andrew Oldham (Camberley, Surrey)

```
VDU 23,233,127,128,191,
160,160,175,162,162

VDU 23,234,255,0,255,
1,1,165,37,37

VDU 23,235,254,65,
85,65,81,65,81,65

VDU 23,236,162,162,162,
162,160,191,128,127

VDU 23,237,37,37,61,
25,1,255,0,255

VDU 23,238,81,65,81,
65,65,93,65,254
```

Umbrella

From Matthew Swallow (Lowdham, Notts)

```
VDU 23,226,24,60,126,
255,16,16,16,16

VDU 23,227,80,112,0,
0,0,0,0,0
```

Devil's Head

From Alan Lowdham (Birmingham)

```
VDU 23,225,66,90,126,
90,255,66,60,24
```

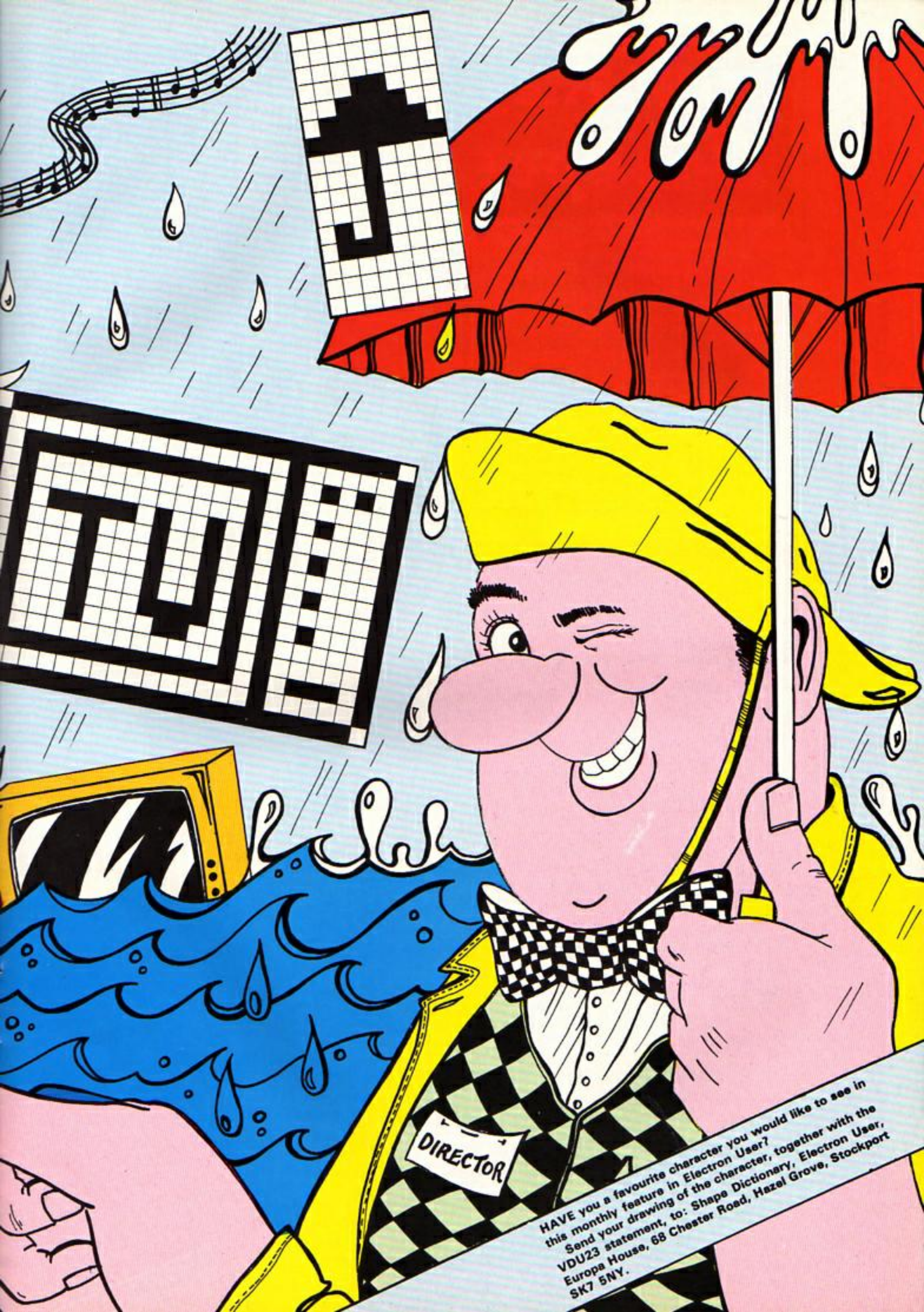
A Quaver

From T. Stewart (East Lothian)

```
VDU 23,239,8,12,
8,8,8,8,24,24
```

A Crotchet

```
VDU 23,240,8,8,8,
8,8,8,24,24
```



HAVE you a favourite character you would like to see in this monthly feature in Electron User?
Send your drawing of the character, together with the VDU23 statement, to: Shape Dictionary, Electron User, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

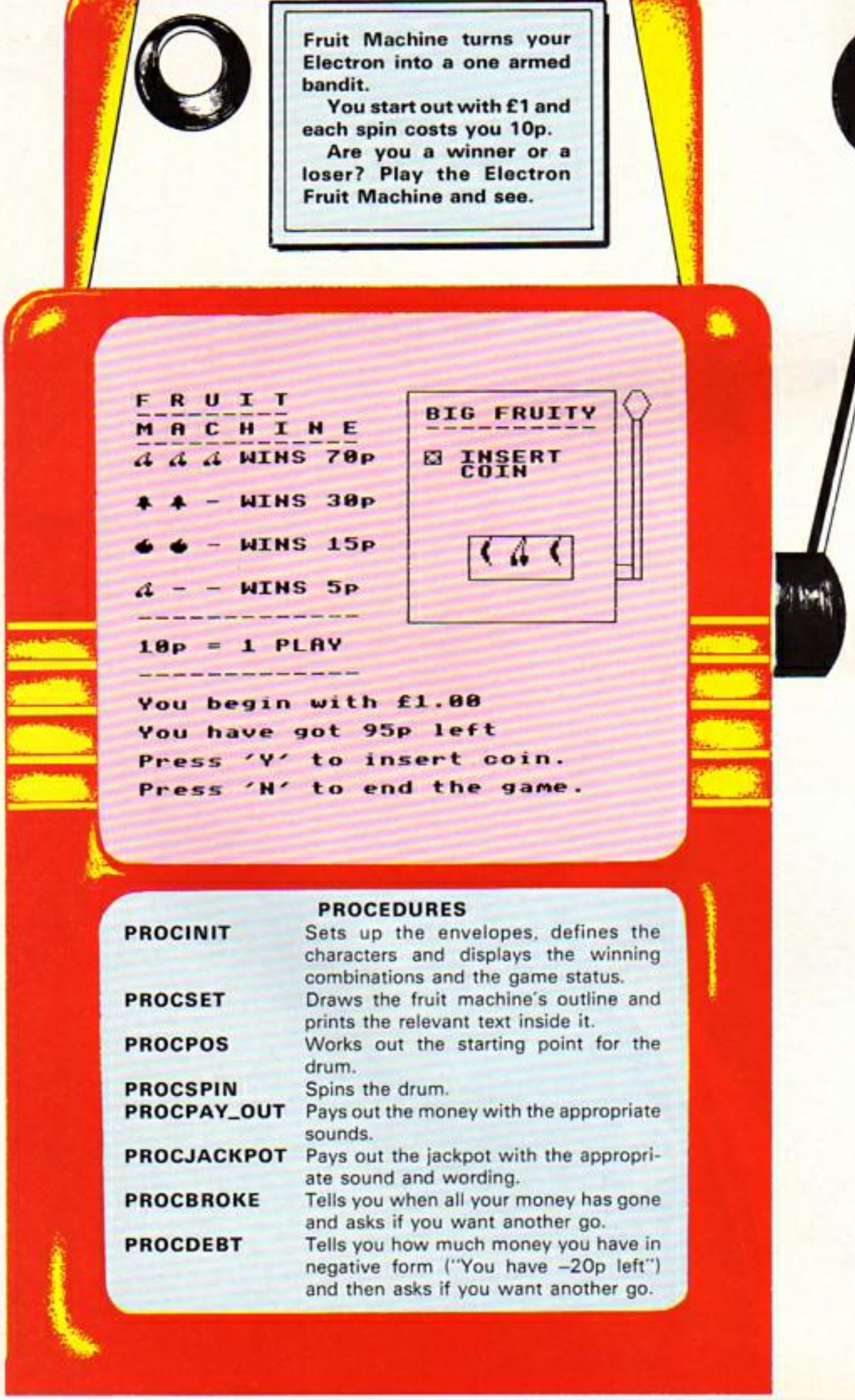
HIT THE Jackpot

By JAMES ROOK

Fruit Machine turns your Electron into a one armed bandit.

You start out with £1 and each spin costs you 10p.

Are you a winner or a loser? Play the Electron Fruit Machine and see.



PROCEDURES

PROCINIT	Sets up the envelopes, defines the characters and displays the winning combinations and the game status.
PROCSET	Draws the fruit machine's outline and prints the relevant text inside it.
PROCPOS	Works out the starting point for the drum.
PROCSPIN	Spins the drum.
PROCPAY_OUT	Pays out the money with the appropriate sounds.
PROCJACKPOT	Pays out the jackpot with the appropriate sound and wording.
PROCBROKE	Tells you when all your money has gone and asks if you want another go.
PROCDEBT	Tells you how much money you have in negative form ("You have –20p left") and then asks if you want another go.

```
 10 REM FRUIT MACHINE
 20 REM BY J.ROOK
 30 REM (C) ELECTRON USER
 40 MODE 4
 50 PROCINIT
 60 M=100
 70 PROCSET
 80 PRINT TAB(23,9);"
        ";TAB(23,10);
    "COIN";TAB(23,11);
    "INSERTED"
 90 PROCPOS
100 M=M-10
110 PROCSPIN
120 PROCPAY OUT
130 IF M<0
    THEN PROCDEBT
140 IF M=0
    THEN PROCBROKE
150 PRINT TAB(25,13);"
      "
160 PRINT TAB(23,19);"
          "
170 PRINT TAB(0,22);"You
    begin with £1.00"
180 PRINT '"You have got
    ";M;"p left";SPC (3)

190 PRINT TAB(23,11);"
          ";TAB(23,9);
    "INSERT";TAB(23,10);
    "COIN"
200 PRINT TAB(0,26);"Press
    'Y' to insert coin."
210 PRINT '"Press 'N' to
    end the game."
220 A$=GET$
230 IF A$="N"
    THEN GOTO 250
240 IF A$="Y"
    THEN SOUND 1,-15,150
    ,5
    :GOTO 70
250 PRINT '"You end the
    game with ";M;"p"
260 END
270 DEF PROCPOS
280 X=(RND(4)-1)*10+1
290 Y=(RND(4)-1)*10+1

300 Z=(RND(4)-1)*10+1

310 ENDPROC
320 DEF PROCSPIN
330 S=RND(2)+2
340 FOR I=0 TO S*10
350 VDU 23,229,C(X),C(X)
    ,C(X+1),C(X+1),C(X+2)
    ,C(X+2),C(X+3),C(X+3)
360 VDU 23,232,C(X+4)
    ,C(X+4),C(X+5),C(X+5)
```

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
    ,C(X+6),C(X+6),C(X+7)
    ,C(X+7)
370 VDU 23,230,C(Y),C(Y)
    ,C(Y+1),C(Y+1),C(Y+2)
    ,C(Y+2),C(Y+3),C(Y+3)

380 VDU 23,233,C(Y+4)
    ,C(Y+4),C(Y+5),C(Y+5)
    ,C(Y+6),C(Y+6),C(Y+7)
    ,C(Y+7)
390 VDU 23,231,C(Z),C(Z)
    ,C(Z+1),C(Z+1),C(Z+2)
    ,C(Z+2),C(Z+3),C(Z+3)

400 VDU 23,234,C(Z+4)
    ,C(Z+4),C(Z+5),C(Z+5)
    ,C(Z+6),C(Z+6),C(Z+7)
    ,C(Z+7)
410 PRINT TAB(24,15);
    CHR$ (229);" ";
    CHR$ (230);" ";
    CHR$ (231)
420 PRINT TAB(24,16);
    CHR$ (232);" ";
    CHR$ (233);" ";
    CHR$ (234)
430 IF X=40
    THEN X=0
440 IF Y=40
    THEN Y=0
450 IF Z=40
    THEN Z=0
460 X=X+1
    :Y=Y+1
    :Z=Z+1
470 NEXT I
480 X=X-1
    :Y=Y-1
    :Z=Z-1
490 ENDPROC
500 DEF PROCPAY_OUT
510 IF X=1 AND Y=1
    AND Z=1
    THEN PROCJACKPOT
    :ENDPROC
520 IF (X=31)+(Y=31)+(Z=31)
    =-2
    THEN M=M+15
    :PRINT TAB(25,13);
    "15p"
    :A$=INKEY$ 300
    :SOUND 1,3,148,22
    :ENDPROC
530 IF (X=21)+(Y=21)+(Z=21)
    =-2
    THEN M=M+30
    :PRINT TAB(25,13);
    "30p"
    :A$=INKEY$ 300
    :SOUND 1,3,148,44
    :ENDPROC
540 IF (X=1)+(Y=1)+(Z=1)=-1

    THEN M=M+5
    :PRINT TAB(26,13);
    "5p"
    :A$=INKEY$ 300
    :SOUND 1,3,148,7
    :ENDPROC
550 ENDPROC
560 DEF PROCINIT
570 ENVELOPE 3,2,-25,-80
    ,-6,15,0,0,126,0,0
    ,-126,126,126
580 ENVELOPE 1,3,-17,61
    ,9,4,0,0,126,0,0,-126
    ,126,126
590 VDU 23,224,&06,&0A
    ,&14,&24,&44,&CF,&EF
    ,&E6
600 VDU 23,225,&02,&0C
    ,&1C,&38,&38,&1C,&0C
    ,&02
610 VDU 23,226,&18,&3C
    ,&3C,&3C,&7E,&FF,&18
    ,&18
620 VDU 23,227,&0C,&18
    ,&7A,&FF,&FF,&FF,&7E
    ,&3C
630 VDU 23,228,&00,&00
    ,&00,&7E,&7E,&00,&00
    ,&00
640 VDU 23,235,255,153
    ,165,195,195,165,153
    ,255
650 VDU 23,1;0;0;0;0
660 PRINT TAB(0,1);"F R
    U I T"
670 PRINT "---------"
680 PRINT "M A C H I N
    E"
690 PRINT "-------------"
700 PRINT TAB(0,5);
    CHR$ (224);" ";
    CHR$ (224);" ";
    CHR$ (224);" WINS 70p"

710 PRINT ''CHR$ (226);
    " ";CHR$ (226);" -
    WINS 30p"
720 PRINT ''CHR$ (227);
    " ";CHR$ (227);" ";
    "- ";"WINS 15p"
730 PRINT ''CHR$ (224);
    " - - WINS 5p"
740 PRINT '"-------------"
750 PRINT '"10p = 1 PLAY"
760 PRINT '"-------------"
770 DATA &06,&0A,&14,&24
    ,&44,&CF,&EF,&E6,0
    ,0
780 DATA &02,&0C,&1C,&38
    ,&38,&1C,&0C,&02,0
    ,0
790 DATA &18,&3C,&3C,&3C
    ,&7E,&FF,&18,&18,0
    ,0
800 DATA &0C,&18,&7A,&FF
    ,&FF,&FF,&7E,&3C,0
    ,0
810 DATA &06,&0A,&14,&24
    ,&44,&CF,&EF,&E6,0
    ,0
820 DIM C(48)
830 FOR I=1 TO 48
840 READ C(I)
850 NEXT I
860 ENDPROC
870 DEF PROCJACKPOT
880 SOUND 1,1,157,40
890 PRINT TAB(23,19);"JACKP
    OT"
900 PRINT TAB(25,13);"70p"
910 A$=INKEY$ 250
920 M=M+70
930 ENDPROC
940 DEF PROCBROKE
950 PRINT TAB(0,24);"You
    have no more money."
960 PRINT '"
                    "
970 PRINT TAB(0,26);"You
    are broke!"
980 PRINT '"
                    "
990 INPUT TAB(0,28);"Do
    you want another go
    ",A$
1000 A$=LEFT$(A$,1)
1010 IF A$="N"
    THEN END
1020 IF A$="Y"
    THEN RUN
1030 ENDPROC
1040 DEF PROCDEBT
1050 PRINT TAB(0,24);"You
    have got ";-(0-M);
    "p left"
1060 PRINT '"
                    "
1070 PRINT TAB(0,26);"You
    are in debt!"
1080 PRINT '"
                    "
1090 INPUT TAB(0,28);"Do
    you want another go
    ",A$
1100 A$=LEFT$(A$,1)
1110 IF A$="N"
    THEN END
1120 IF A$="Y"
    THEN RUN
1130 ENDPROC
1140 DEF PROCSET
1150 PRINT TAB(0,24);"

    "
1160 PRINT '"
                    "
1170 PRINT '"
                    "
1180 MOVE 640,352
    :DRAW 640,864
    :DRAW 1024,864
    :DRAW 1024,352
1190 DRAW 640,352
    :MOVE 1024,448
    :DRAW 1072,448
    :DRAW 1072,800
1200 DRAW 1088,832
    :DRAW 1072,864
    :DRAW 1056,864
    :DRAW 1040,832
1210 DRAW 1056,800
    :DRAW 1072,800
    :MOVE 1056,800
    :DRAW 1056,448
1220 MOVE 1056,480
    :DRAW 1024,480
    :MOVE 752,448
    :DRAW 752,544
1230 DRAW 944,544
    :DRAW 944,448
    :DRAW 752,448
1240 PRINT TAB(21,6);"BIG
    FRUITY"
1250 PRINT TAB(21,7);"------
    ----"
1260 PRINT TAB(21,9);
    CHR$ 235
1270 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 47

Look out for ROMs.

IN the first of the series *(Electron User,* March 1984) we looked at what specifies a memory location. Most memory locations address either RAM or ROM, so this month we'll talk about exactly what they are.

Each memory location contains eight bits, or one byte, of information. We need two types of memory as they each have different properties.

ROM stands for Read Only Memory. The address locations that contain ROM cannot be altered by the computer. They are fixed at the time the chip is made.

The designers of the ROM have to give the manufacturers a tape of the required contents. Then the bit pattern for each address is built into the ROM chip.

This makes them very expensive initially, as the manufacturers incur a lot of expense tooling up.

But if the chips are to be made in any quantity the tooling charge amounts to a small fraction of the total cost of the device.

ROM is a very English type of memory – that is, it is non-volatile.

In other words it keeps its head and will retain what is placed in it even when the power is removed.

This makes it ideal for storing programs and data which have to be instantly available every time the computer is switched on.

In the Electron this means the program which allows the computer to understand Basic program statements as well as data such as the shape of the letters and numbers you see on the screen.

The ROM in the Electron takes up 32k of address space – that is half the total available memory.

The advantage of this is that the version of Basic it contains is very powerful and so you don't need so much room for your program as you would on other machines.

The rival Spectrum is advertised as having a massive memory, but in fact it is no bigger than the Electron's.

The only difference between them is the proportion of ROM to RAM.

RAM stands for Random Access Memory, which makes life a little confusing at first as ROM memory can also be accessed randomly. It means that address locations can be read in any order.

The name is really a hangover from the days when all forms of read/write memory could only be accessed sequentially.

The contents of this memory were constantly circulating and you had to wait before the required address appeared through a small "window" before you could get it.

This was in the days of large mainframe computers which only ran batch programs from punched cards. They were not interactive like the Electron.

So you see that the word RAM describes the sort of memory which can be altered by the user.

RAM is volatile. That is, when the power is removed it will forget whatever was stored in it.

As it is used to store your programs, you'll see that they will need to be loaded in every time the Electron is switched on.

This is why there is a tape recorder output on most computers to enable programs to be loaded rapidly.

When RAM is first powered up, it will contain a collection of zeros and ones known as "rubbish". This is because they have no significance and the contents cannot be predicted.

It is true that any individual device tends to power up with the same rubbish every time and with different devices the rubbish is different.

The Electron usually clears out most of its memory on power-up. But the memory which contains the single letter integer variables, such as *A%, B%* is left untouched. On switch-on these will contain rubbish.

Due to the nature of the type of memory used you will tend to get the value of zero (all memory locations logic zero) or minus one (all memory locations one).

Try printing these out just after switch-on, but remember they are likely to be different on other Electrons.

There are two types of RAM – static and dynamic.

Static RAM can be thought of as consisting of a number of buckets, either containing

.. and RAMs

WHAT'S THAT WHEN IT'S AT HOME?

MIKE COOK explains the inner workings of the Electron Part 2

water or empty (logic one or zero).

The buckets can be looked into to see what they contain when the memory location is read. When the power is switched off, they all get joggled about and the water sloshes all over the place.

Now dynamic RAM, the type in the Electron, works in the same way except that the buckets are leaky, and they have to be constantly looked at and topped up if necessary.

This has to be done about 500 times a second and is known as refreshing.

Fortunately lots of buckets can be refreshed at the same time and so only 128 locations need to be refreshed.

Even so, each one of these must still be refreshed 500 times a second.

If the memory were not refreshed its contents would "leak" away.

You may ask: "Why bother with dynamic RAM with its need to be constantly refreshed when static RAM is available?"

Well if you think about it, it takes less material to make a leaky bucket than it does to make a sound one. Exactly the same applies for RAM.

There are fewer components making up dynamic RAM, so you can pack more of them onto a chip.

As the cost of making a chip is the same whether it contains static or dynamic RAM, you get more memory for your money.

In addition, dynamic RAM uses less electricity, and so it does not get as hot or need such big power supplies.

If you are going to have more than just a few k of memory it makes economic sense to use dynamic RAM.

Now the Electron uses the latest type of dynamic RAM with 64k address locations in each package.

Each package contains one bit of information, so to get a byte you would need eight packages.

However that would give you 64k of memory and as we have already said, the ROM occupies 32k of the total 64k maximum memory locations.

To get round this the Electron has only four packages of RAM, and each one is accessed twice to build up one byte.

Extra electronics look after this so that the computer is not aware it is happening.

The only snag is that as you have to look at the RAM twice to get a byte it takes twice as long.

The computer's overall speed is governed by how fast the memory can be read. This is known as the memory access time.

Only RAM memory accesses are slowed down in this way. ROM is read at full speed, which explains why the Electron is only about 40 per cent slower than the BBC Micro.

You can only say "about" because this depends upon the proportion of RAM to ROM the computer is accessing, and this will depend on the program being executed.

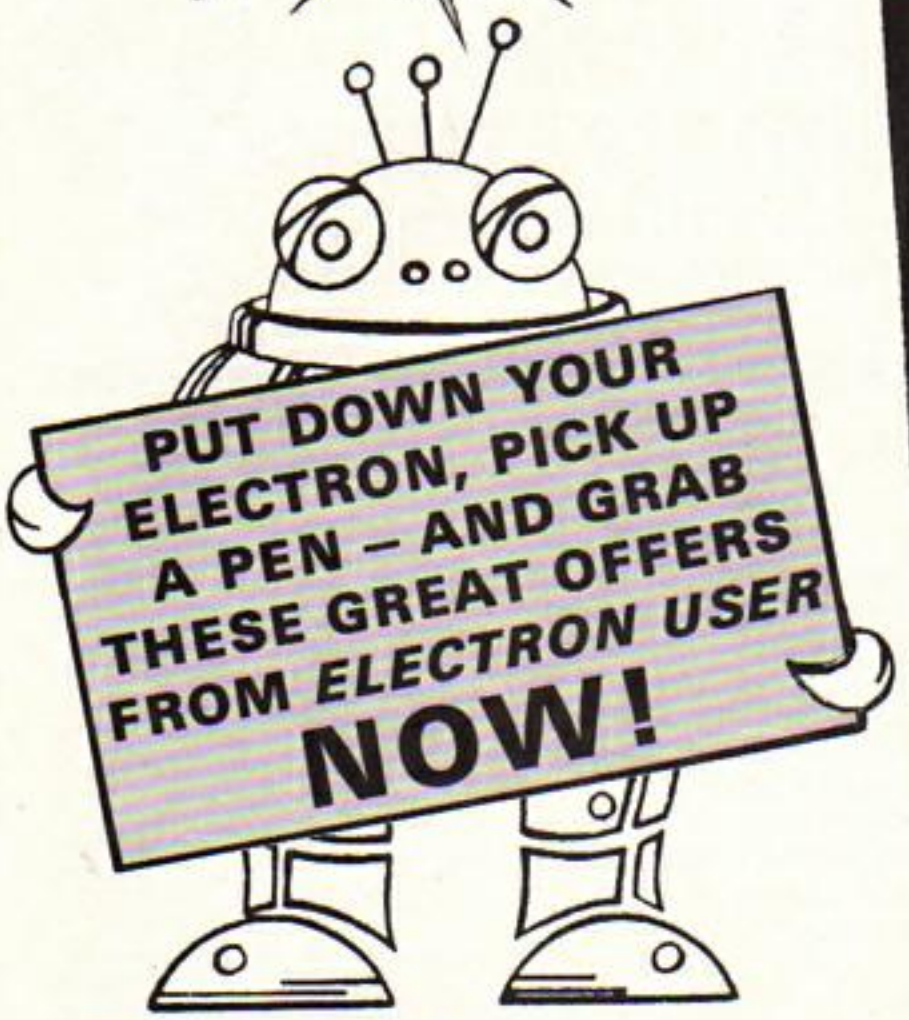
So there you have it – two types of memory: ROM for keeping permanent information, and RAM for keeping temporary or changing information.

There is a device which combines the best of these two called non-volatile RAM. But at the moment it is only available with very small capacities and is very expensive.

No doubt it will be incorporated into most computers, but not, I suspect, for the next 10 years.

Next time we will be looking at the microprocessor itself and seeing exactly what it does.

electron user

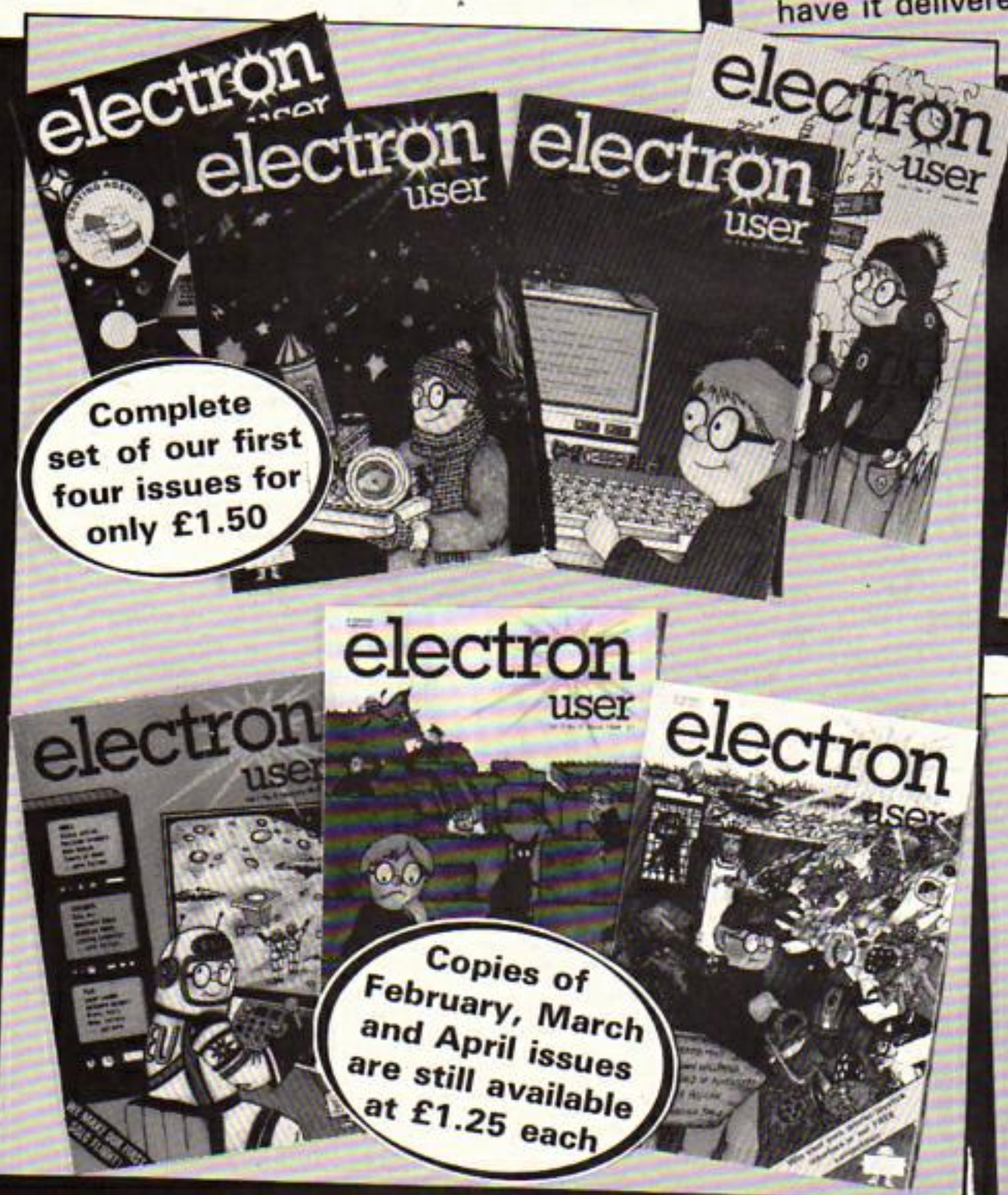


Be one of the first to get each issue

A subscription will ensure you get your own personal copy HOT OFF THE PRESSES month after month for the next year.

Every owner of an Electron – and everyone thinking of buying one – needs to get Electron User every month. It's the brightest, most authoritative yet completely independent guide to a machine that has so much potential you will never tire of reading about its remarkable capabilities.

You can buy Electron User from your local newsagent or station bookstall. Or you can take out a 12 months subscription and have it delivered to you by post.



Your Electron needs protecting!

Protect your Electron with our luxury dust cover made of soft pliable water-resistant vinyl, bound with strong cotton and decorated with *Electron User* logo.

£3.95

Keep your collection of *Electron User* complete with these handsome binders

Bound in attractive red pvc with the *Electron User* logo in gold blocking on the spine, this binder will hold 12 magazines firmly secured in place by metal rods. **£3.95**

FREE ***Cassette worth £3.75 if you subscribe NOW!***

If you take out a subscription to Electron User now you will receive completely free one of the monthly cassettes of Electron User listings. Choose which one you want from those illustrated below.

This free gift is for a limited period, so subscribe now!

Cassette tapes of *Electron User* programs

Save typing in programs from *Electron User* by sending for these program-packed tapes.

£3.75 each

You can also take out a subscription for the 12 monthly tapes for £40.

ORDER FORM

All prices include postage, packing and VAT, and are valid to May 25.

Please enter number required in box — £ p

Electron User annual subscription

UK £12 ☐
EIRE £13 (IR £16) ☐
Overseas (Surface) £20 ☐
Overseas (Airmail) £40 ☐

Selected free cassette _____ (month)
Commence with _____ issue TOTAL _____

Electron User introductory issues

£1.50 UK
£1.75 Overseas (Surface)

Complete set of 4 ☐

TOTAL _____

Electron User back issues

£1.25 UK
£1.50 Overseas (Surface)
Airmail prices on application

February ☐
March ☐
April ☐

TOTAL _____

Electron User tapes

£3.75 (UK & Overseas)

26 introductory programs ☐
Lunar Lander. February ☐
Chicken. March ☐
Spacehike. April ☐
Rally Driver May ☐

TOTAL _____

Cassette tape annual subscription

£40 (UK & Overseas) ☐
Commence with _____ tape (state month) TOTAL _____

Dust Cover

£3.95 (UK & Overseas) ☐

TOTAL _____

Binder

£3.95 UK
£5.00 Overseas ☐

TOTAL _____

Payment: please indicate method (✓) **TOTAL**

☐ Access/Mastercharge/Eurocard
☐ Barclaycard/Visa
☐ American Express

Card No. _____

Expiry Date _____

☐ Cheque/PO made payable to Database Publications Ltd

Name _____

Address _____

_____

Signed _____

Send to: *Electron User*, FREEPOST, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

(No stamp needed if posted in UK) *Please allows 28 days for delivery*

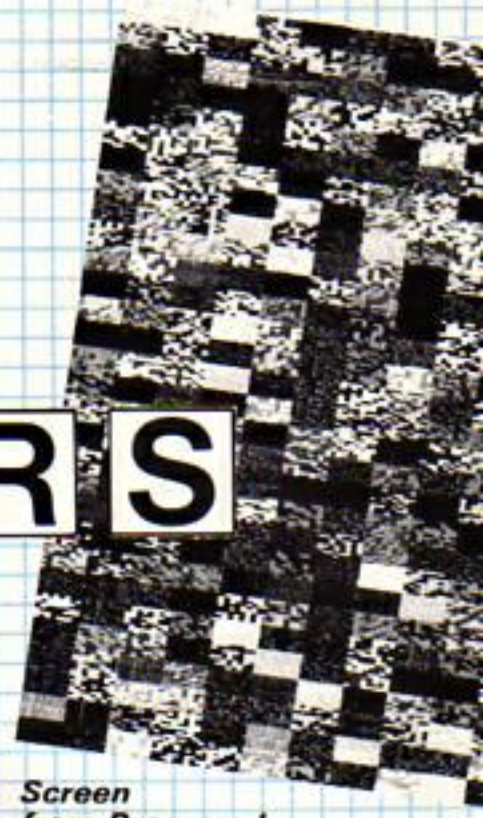
You can also order by phone

Telephone: 061-480 0171
24 hours

Don't forget to quote your credit card number and give your full address

Get yourself involved with PETER GREY's

SHADY CHARACTERS



Screen from Program I

HAVE you ever had a go at designing user-defined characters like the ones that appear in Casting Agency each month?

They take quite a lot of thought, effort, and planning with pencil and paper.

Being fairly lazy, I wondered if there was some way I could get my Electron to do the work for me and create some user-defined characters.

The three programs in this article came from that idle speculation.

If you look at Figure I you'll see that it's the usual way of calculating a user-defined character. You add up each of the rows and get a number between 0 and 255.

"Why not", I asked myself, "use the RND function to produce a random number for each line?"

Each character has eight lines making it up, so if we get the Electron to generate eight random numbers and put them behind a VDU 23, the result is a purely random user-defined character.

This is what PROCcharacter does in Program I. It generates eight random numbers and stores them in an array.

PROCshade picks a random foreground and background colour for the character, making sure that the two are not the same.

The program runs in Mode 2 and the FOR . . . NEXT loop calls the two procedures over and over, filling the screen with coloured, random characters.

The trouble is that the bottom line scrolls up, spoiling the effect.

I wondered if I could do the same sort of thing using the graphics command GCOL instead of PRINTing the character each time. I came up with Program II.

This uses the same two procedures, only PROCshade is but a shadow of its former self!

The GCOL 0 that I use just prints the foreground colour, so I left the background as black.

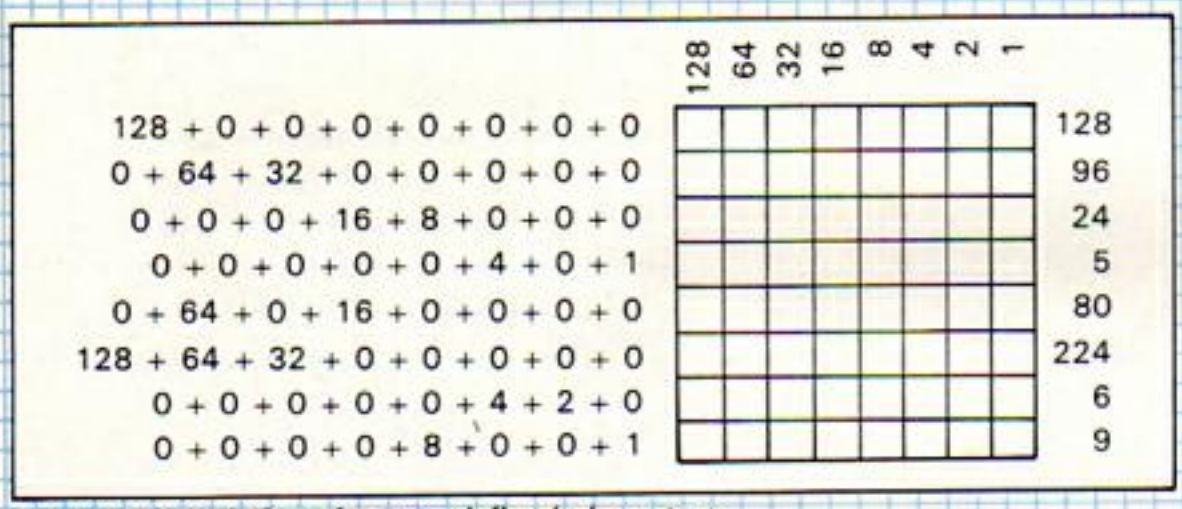


Figure I: Calculations for user-defined characters

The program works very much as before, a FOR . . . NEXT loop filling the screen with the randomly shaped and coloured characters.

The difference is that now graphics commands are used, so the loop parameters have different values.

Again I wasn't satisfied. The black background was all very nice, but couldn't I have a multicoloured background like I had in Program I?

I would have to print at exactly the same spot twice, once in one colour, then again in a second.

Also, as I was using GCOL, the colours mustn't overlap. This meant that the user defined (or, rather, Electron defined) characters had to be the reverse of each other.

One must have its foreground colour where the other has its background colour and vice versa.

Program III shows the result of my deliberations.

It's very similar to Program II, only now PROCshade has some of its former glory, picking two colours.

PROCcharacter defines two characters, the second being the opposite of the first.

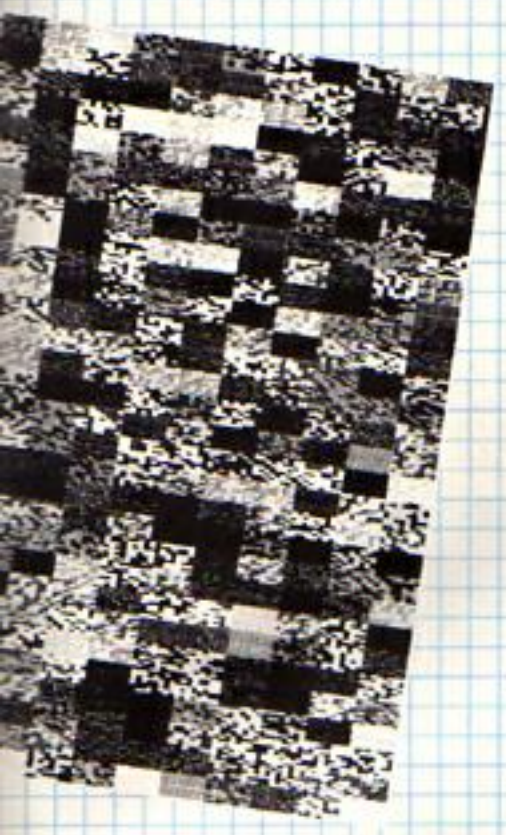
The main program prints the two together at the same spot on the screen. Figure II shows how it's done.

Those are the three programs that came from being too idle to create my own characters.

The patterns are nice, but it

```
10 REM PROGRAM I
20 MODE 2
30 DIM byte(8)
40 FOR row=0 TO 19
50 FOR line=0 TO 30
60 PROCshades
70 PROCcharacter
80 COLOUR colour
90 COLOUR backcolour
100 PRINT TAB(row,line)
    CHR$ (224)
110 NEXT line
120 NEXT row
130 END
140 DEF PROCcharacter
150 FOR generator=1 TO 8
160 byte(generator)=RND(256)-1
170 NEXT generator
180 VDU 23,224,byte(1),byte(2
    ),byte(3),byte(4),byte(5)
    ,byte(6),byte(7),byte(8)
190 ENDPROC
200 DEF PROCshades
210 colour=RND(8)-1
220 backcolour=RND(8)+127
230 IF colour=backcolour
    THEN PROCshades
240 ENDPROC
```

```
10 REM PROGRAM II
20 MODE 2
30 VDU 5
40 DIM byte(8)
50 FOR row=0 TO 1279
   STEP 64
60 FOR line=0 TO 1023
   STEP 32
70 PROCshades
80 PROCcharacter
90 GCOL 0,colour
100 MOVE row,line
   :VDU 224
110 NEXT line
120 NEXT row
130 END
140 DEF PROCcharacter
150 FOR generator=1 TO 8
160 byte(generator)=RND(256)-1
170 NEXT generator
180 VDU 23,224,byte(1),byte(2
    ),byte(3),byte(4),byte(5)
    ,byte(6),byte(7),byte(8)
190 ENDPROC
200 DEF PROCshades
210 colour=RND(7)
220 ENDPROC
```



was the programming that gave the satisfaction. And it isn't finished yet.

I wonder how I can get the final pattern to flash?

It'll be something using random VDU19s in a REPEAT... UNTIL loop. The trick will be avoiding using the same colour for foreground and background.

I wonder...

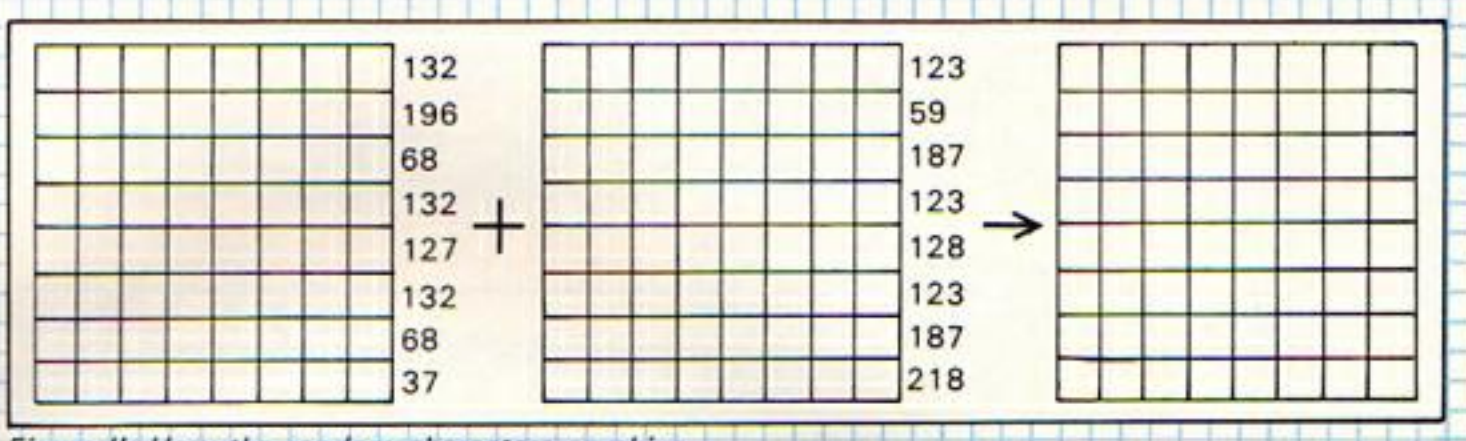
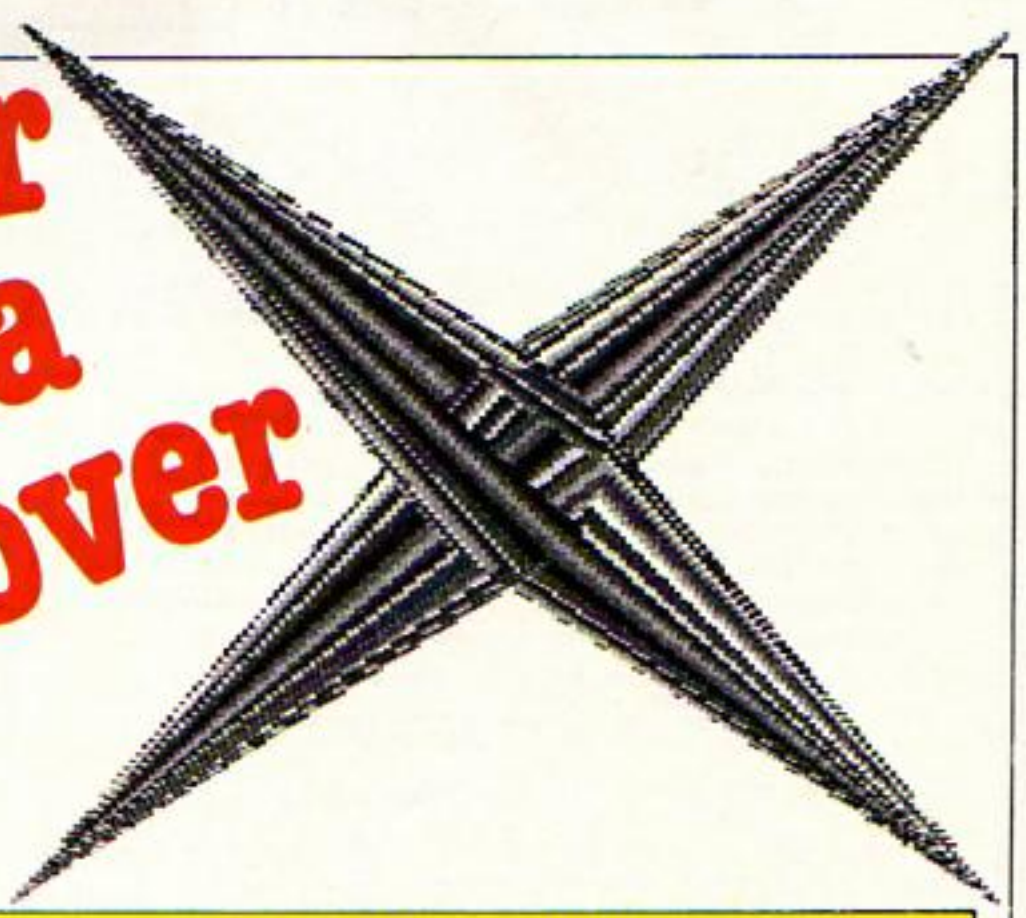


Figure II: How the random characters combine

```
10 REM PROGRAM III
20 MODE 2
30 VDU 5
40 DIM byte(8)
50 FOR row=0 TO 1279
   STEP 64
60 FOR line=0 TO 1023
   STEP 32
70 PROCshades
80 PROCcharacter
90 GCOL 0,colour1
100 MOVE row,line
   :VDU 224
110 GCOL 0,colour2
120 VDU 8,225
130 NEXT line
140 NEXT row
150 END
160 DEF PROCcharacter
170 FOR generator=1 TO 8
180 byte(generator)=
   RND(256)-1
190 NEXT generator
200 VDU 23,224,byte(1)
   ,byte(2),byte(3),byte(4
   ),byte(5),byte(6)
   ,byte(7),byte(8)
210 VDU 23,225,255-byte(1)
   ,255-byte(2),255-byte(3
   ),255-byte(4),255-byte(
   5),255-byte(6),255-byte
   (7),255-byte(8)
220 ENDPROC
230 DEF PROCshades
240 colour1=RND(8)-1
250 REPEAT
   :colour2=RND(8)-1
   :UNTIL colour2<>colour1
260 ENDPROC
```

Make your Electron a lovely mover



MAKE your Electron really move with Mark Smiddy's short but powerful program, Rotation.

```
10 REM ROTATE
20 REM BY MARK SMIDDY
30 REM (C) ELECTRON USER
40 MODE 2
50 PROCdraw(640,512)
60 PROCdraw(640,-512)
70 PROCdraw(-640,-512)
80 PROCdraw(-640,512)
90 PROCdraw(0,0)
100 PROCrot
110 DEF PROCdraw(X%,Y%)
120 VDU 29,640;512;
   :MOVE X%,Y%
   :N%=0
   :R%=120
130 FOR I = 0 TO 2*PI
   STEP 0.1
140 GCOL 0,N%
   :N%=(N%+1)MOD 15
150 MOVE X%,Y%
160 PLOT 85,R%*COS I,R%*
   SIN I
170 NEXT
180 ENDPROC
190 DEF PROCrot
200 VDU 20
210 REPEAT
220 D%=0
230 FOR D%=1TO 7
240 FOR C%=1TO 15
250 FOR N=0TO 40
   :NEXT
260 VDU 19,C%,D%;0;
270 NEXT:NEXT
280 UNTIL 0
290 ENDPROC
```

MRM SOFTWARE

DIAMOND MINE

An absolutely fascinating and compulsive game designed to test your skills of dexterity and coordination. Steer the lengthening pipe to the diamonds buried deep in the depth of the diamond mine. But be careful avoid hitting the walls with the pipe and watch out for those monstrous menacing meanies . . . the bugs.

Have you enough pipe left to collect all the diamonds? Have you got steady hands? Find out with this sparkling gem from MRM.

ONLY £5.70
Enclose 55p p&p per order

"Simple game, but quite a find for all that".
"This is one of those surprising little games that always manages to tempt you back for another go".
Computer Games, March 1984

MRM SOFTWARE

MRM SOFTWARE
17 Cross Coates Road
GRIMSBY
South Humberside
Telephone: 0472 44304

Dealers telephone now for dealer prices

Software Surgery

THE COLUMN THAT TAKES A LOOK INSIDE THE LATEST RELEASES

Great way to experiment with graphics

ELECTRON GRAPHICS
Salamander Software

THE Electron is capable of supporting a wide range of graphics and text modes, better in fact than many machines costing much more.

Imaginative programming can be carried out in Modes 0, 1 and 2, although many people would find the GCOL, MOVE, DRAW and PLOT statements difficult to plan for an involved drawing.

This program takes the difficulty away, substituting it with a series of simple commands with which complicated, colourful and concise artwork can be designed.

Only Modes 0, 1 and 2 can be used, and the available colours are shown on a palette at the bottom of the screen.

Should other colours than the default one be required it is simple to alter those available.

A flashing cross-hair cursor is used to position elements, and the coordinates are constantly updated on-screen.

A number of built-in functions can be used, and each has an easily remembered mnemonic. B draws a box, C sets a circle, F fulfills a FILL function, L produces a line while A initiates an arc.

For all these, when the cursor is in the correct position, the Spacebar is the input necessary to start the procedure.

Text can be added at will on the screen, and so many applications spring to mind.

Pie charts and histograms may be labelled and coloured to relay information, systems may be designed, and complicated maps and drawings transferred from graph paper.

Pictures may be built up in a series of pages and may be stored onto cassette for future use.

One glaring omission, looking to the future, is that there seems to be no facility for a screen dump.

A hard copy of the screen display would be a fitting final facility to this useful piece of software.

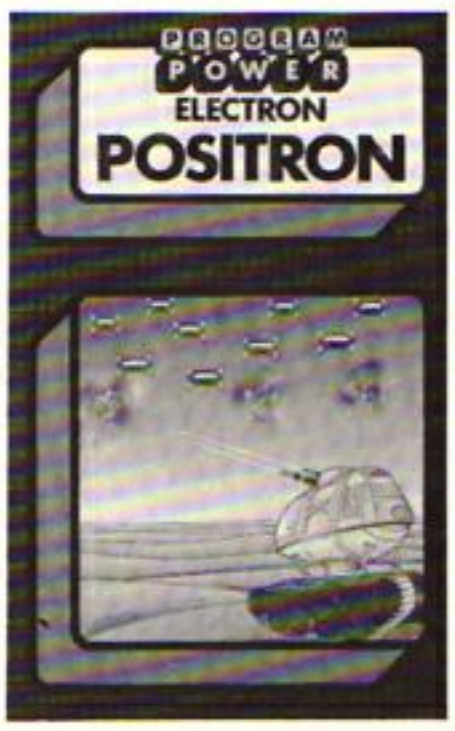
It fulfills a large variety of purposes, and also stands on its own as great fun with which to experiment.

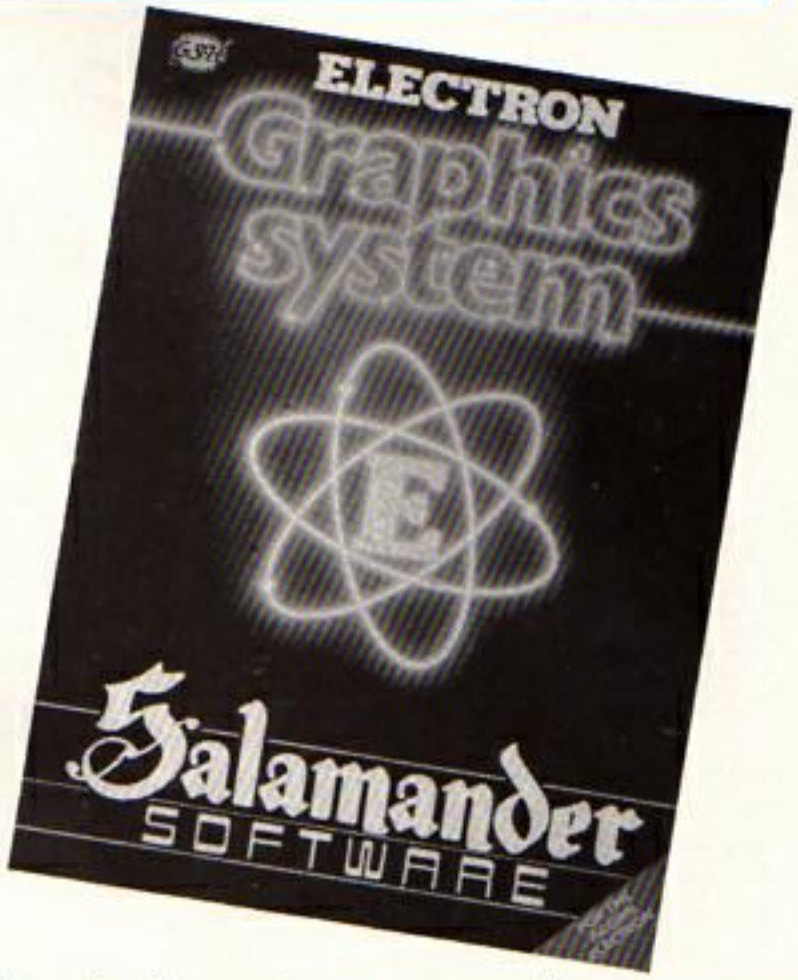
Phil Tayler

Fast mover

POSITRON
Program Power

YOU'VE seen it all before. The space invaders tramp predictably across the screen, edging relentlessly earthwards.





You wipe them out mercilessly with your quick firing laser base, rapidly clearing the first screen.

It appears all too easy – then all hell breaks loose . . .

The second wave doesn't follow a set pattern. They swarm about all over the place setting up defensive boxes. If you don't break them up they will be your downfall.

Moving quickly earthwards they have landed before you can gather your shattered wits.

And that's only the second wave – there are nine in all, each progressively worse.

Positron is a fast moving, colourful and satisfying game. So sharpen your wits, tighten your sweatband and give it a whirl.

Adam Young

Learning can be fun

10 EDUCATIONAL GAMES
Dimax Structured Software

WHENEVER I see a compendium tape my reaction is to shudder. There is usually one reasonable program surrounded by a heap of others which vary from bad to awful.

This one, however, is a pleasant exception, containing 10 programs aimed at the user in school.

One superb innovation is that Dimax makes the listing freely available. Each program uses the same standard programming format so that the listing can be adapted to suit special needs.

Indeed, Dimax will even sell the listings separately for just 40p each should your typing not be up to Olympic standard.

The games cover mathematical themes, letter recognition and a stiff test on capital cities. There are also quite reasonable versions of standard games such as Mastermind and Simon.

While none of the ideas is especially original, the versions are well enough programmed to give interesting screen displays.

There is even a version of Tree of Knowledge, a simple introduction to the setting up of a datafile.

For less than the price of a normal commercial program this tape offers a wide range of

→

From Page 51

educational games which can be freely adapted. Indeed, there are even suggestions printed on the inlay of ideas to try.

Many parents will also find this a worthwhile purchase,

especially as it has been written to run on both the Electron and the BBC Micro.

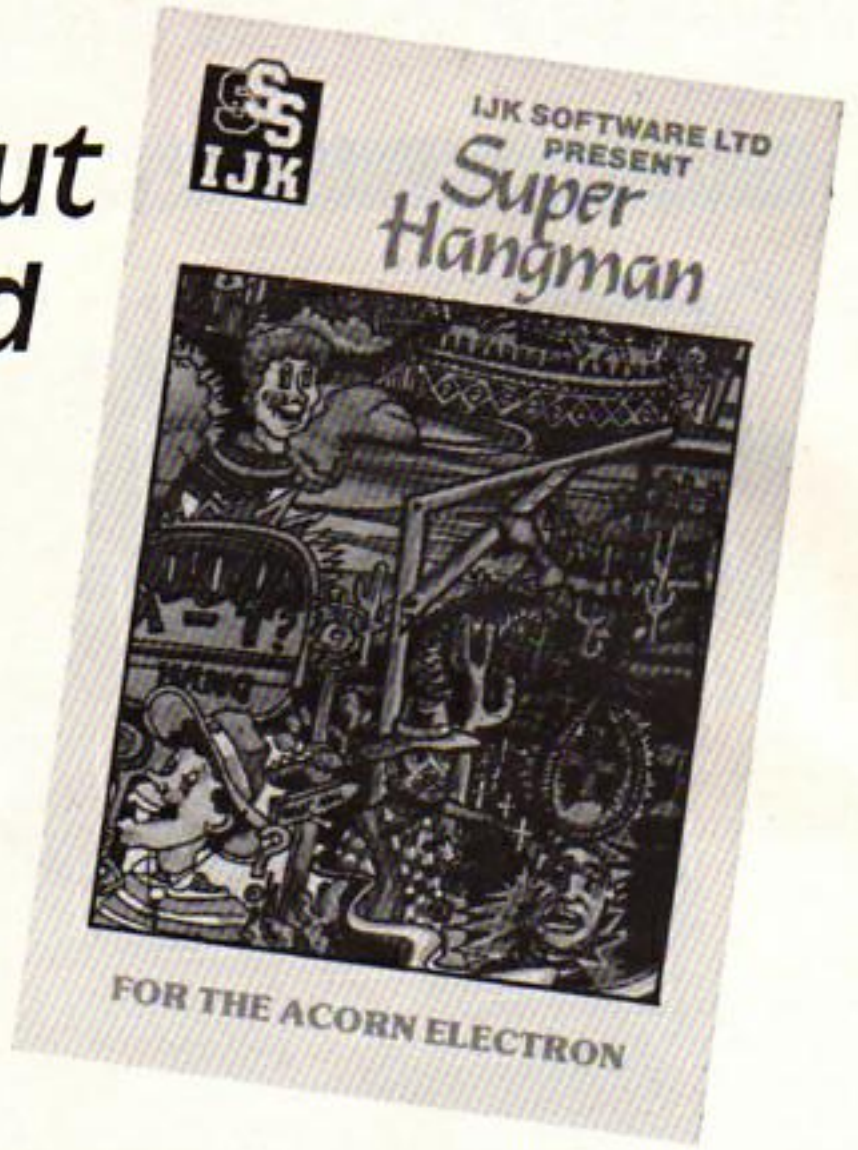
My main criticism is that the Electron is a sophisticated machine which can use colour, detail, sound and animation to stimulate children using the machine.

I am not convinced that Max Lang has exploited this to the full.

Philip Tayler

Hang about – it's an old favourite

HANGMAN
IJK Software



WELL, they did it with Battleships, Gomuku, Othello and even Chess. So why shouldn't they put a really professional version of Hangman on the Electron?

IJK Software have taken this pencil and paper game and turned it into a highly enjoyable video pastime which can be enjoyed by all the family – something rare in video games today.

The graphics are excellent, especially the hanging man, and also colourful. It all adds to the enjoyment of the game.

They have gone into great detail with the victim. He blinks, smiles and clicks his fingers. And if you take too long in contemplating your next choice of letter he will give you quite a surprise.

Forseeing the time when you have learned all the names in each category, there is a section where you can include words of your own choice.

All in all, a simple, good value down-to-earth game and a refreshing change from a screenful of laserbolts and gore.

Adam Young

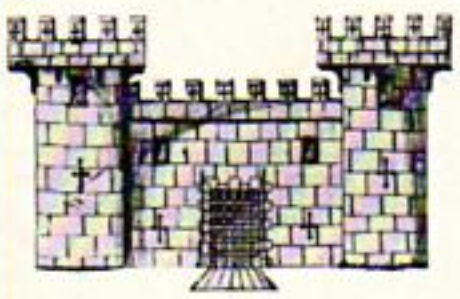
You'll need a lot of bottle

KINGDOM OF KLEIN
Epic Software

THIS is the latest in a series of text-only adventures for the Electron from Epic.

The plot concerns the wicked witch of the mountain who has stolen the Klein bottle from its pedestal in the king's palace.

She has sworn to lay a hideous curse on anyone foolish enough to try to recover it. And the hapless citizens of Klein have elected you to be that fool!



Your task is to find and kill the witch and return the bottle to the pedestal.

You start at the scene of the crime and after collecting some useful items in the palace set out on your quest.

You have a limited amount of movement before encountering the first puzzle – how to cross the river. But having solved this, off you go to the main body of the adventure.

I won't reveal anything else about the game itself. Suffice to say you will meet a belligerent giant, learn to fly and end up in an endless maze.

I consider this a fairly hard adventure, and I must confess I decided to cheat. Imagine my surprise, however, when I found a message in the memory to the effect that "peeking won't help, you'll have to do it the hard way!"

Thus chastened, I returned to the adventure, got a little further and got stuck . . . again.

This time I wouldn't be beaten. A slight alteration to the Ascii values in my dissassembler produced a keywords listing. Thus armed, I hastened back to the game and promptly got stuck yet again.

As I write this, I am finally near the end – the adventures and mine. The effort has been worthwhile, even though I now have a few grey hairs.

Overall a definite must for the experienced adventurer, though the beginner would probably do better with the first of the Epic adventures, Castle Frankenstein.

The save-game facility and response to keyboard input are both very fast.

Electron User index of software reviewers

Castle Frankenstein (Epic Software)	Apr 1984
Caterpillar (IJK Software)	Apr 1984
Croaker (Program Power)	Apr 1984
Cybertron Mission (Program Power)	Apr 1984
Cylon Attack (A&F Software)	Jan 1984
Draughts & Reversi (Acornsoft)	Oct 1983
Draw (Micro Power)	Feb 1984
Electron Chess (Program Power)	Mar 1984
Felix in the Factory (Program Power)	Jan 1984
Grunley Grammar Ghosts (Magic Software)	Dec 1983
Horoscopes (Third Program)	Jan 1984
Kamakazi (A&F Software)	Apr 1984
Meteors (Acornsoft)	Oct 1983
Monsters (Acornsoft)	Oct 1983
Moon Raider (Program Power)	Mar 1984
Pharaohs Tomb (A&F Software)	Mar 1984
Puncman (Chalksoft)	Feb 1984
Starship Command (Acornsoft)	Oct 1983
Supergolf (Squirrel Software)	Mar 1984
Swoop (Program Power)	Dec 1983
Tree of Knowledge (Acornsoft)	Dec 1983
What Makes You Tick? (Third Program)	Feb 1984

A lot of mapping is required and although the solutions to the problems are reasonably easy, finding what you need to solve the problem with can be a headache.

An extremely good adventure and excellent value for money. Recommended.

Merlin

Tic-Tac-Toe listing

From Page 38

```
     A DRAW"
    :SOUND &11,1,120,20
290 PROCcomputer
300 IF (L%=0 AND W%=0)
    PROCpieces
310 IF FNtry("O")COLOUR 3
    :PRINT TAB(0,18)"I win"
    :L%=1
    :SOUND &11,2,20,20
320 UNTIL D%=0 OR W%
    OR L%
330 FOR N=0TO 2000
    :NEXT
    :SOUND &11,0,0,0
340 PRINT TAB(5,30)"PRESS
     SPACE FOR A NEW GAME"
350 REPEAT UNTIL 32=GET
360 UNTIL 0
370 DEF PROCpieces
380 RESTORE 1530
390 FOR N%=1TO 9
    :READ X%,Y%
400 IF Z$(N%)="." COLOUR 7
    :COLOUR 128
    :PRINT TAB(X%,Y%);N%
410 IF Z$(N%)="O" COLOUR 131
    :COLOUR 1
    :PRINT TAB(X%,Y%);"O"
    :SOUND &10,-15,6,1
420 IF Z$(N%)="X" COLOUR 128
    :COLOUR 2
    :
    :PRINT TAB(X%,Y%);"X"
    :SOUND &10,-15,5,1
430 NEXT
440 COLOUR 128
450 ENDPROC
460 DEF PROCboard
470 GCOL 0,1
    :VDU 19,3,4;0;
480 FOR X%=446TO 836STEP 128
490 MOVE X%,320
    :DRAW X%,702
500 NEXT
510 FOR Y%=318 TO Y%+386
     STEP 128
520 MOVE 446,Y%
    :DRAW 830,Y%
530 NEXT
540 ENDPROC
550 DEF PROCplayer
560 IF W% OR L% ENDPROC
570 REPEAT
580 REPEAT COLOUR 3
    :PRINT TAB(13,30)"SQUARE
     NUMBER    ";
    :VDU 127,127,127
590 X%=GET -48
600 UNTIL X%>0 AND X%(=9
610 IF Z$(X%)<>"."VDU 7
620 UNTIL Z$(X%)="."
630 Z$(X%)="X"
640 ENDPROC
650 DEF PROCcomputer
660 IF W% OR L% OR D%=0
    ENDPROC
670 READ X,Y
680 IF rnd=TRUE
    THEN PROCrnd
    :ENDPROC
690 RESTORE 1560
    :T=1
700 REPEAT
710 READ X,Y,Z
720 IF Z$(X)="X" AND Z$(Y)=
    "X"AND Z$(Z)="."
    THEN T=0
730 UNTIL X=0 OR T=0
740 IF T=0
    THEN Z$(Z)="O"
    :ENDPROC
750 RESTORE
760 REPEAT
770 READ X,Y,Z
780 win=FNtest(X,Y,Z,"O")
790 UNTIL X=0 OR win
800 IF win PROCx
    :ENDPROC
810 RESTORE 1520
    :T=1
820 REPEAT
830 READ X,Z,Y
840 IF Z$(X)="X" AND Z$(Y)=
    "X"AND Z$(Z)="."
    THEN T=0
850 UNTIL Z=0 OR X=0
860 IF Z=0 AND Z$(5)="."
    THEN Z$(5)="O"
    :ENDPROC
870 RESTORE
880 REPEAT
890 READ X,Y,Z
900 blk=FNtest(X,Y,Z,"X")
910 UNTIL X=0 OR blk
920 VDU 7
930 IF blk PROCx
    :ENDPROC
940 RESTORE 1540
950 REPEAT
960 READ X,Y
970 IF Z$(X)="X" AND Z$(Y)=
    "."
    THEN X=5
980 UNTIL X=5 OR X=0
990 IF X=5 Z$(Y)="O"
    :ENDPROC
1000 IF Z$(5)="X" P%=-1
    :M%=9
    :
    ELSE P%=0
    :M%=8
1010 REPEAT
    :P%=P%+2
1020 UNTIL Z$(P%)="."
    OR P%>=M%
1030 IF Z$(P%)="."
    THEN Z$(P%)="O"
    :ENDPROC
1040 IF Z$(5)<>"X" P%=0
    :M%=8
    :
    ELSE P%=-1
    :M%=9
1050 REPEAT
    :P%=P%+2
1060 UNTIL Z$(P%)="."
    OR P%>=M%
1070 IF Z$(P%)="."
    THEN Z$(P%)="O"
1080 ENDPROC
1090 DEF PROCrnd
1100 IF Z$(5)="."
    THEN Z$(5)="O"
    :ENDPROC
1110 REPEAT
1120 X%=RND(9)
1130 UNTIL Z$(X%)="."
1140 Z$(X%)="O"
1150 ENDPROC
1160 DEF PROCx
1170 IF Z$(X)="." Z$(X)="O"
    :ENDPROC
1180 IF Z$(Y)="." Z$(Y)="O"
    :ENDPROC
1190 IF Z$(Z)="." Z$(Z)="O"
    :ENDPROC
1200 DEF PROCset
1210 FOR N%=1TO 9
    :Z$(N%)="."
    :NEXT
    :D%=0
    :W%=0
    :L%=0
1220 ENDPROC
1230 DEF PROCins
1240 COLOUR 1
1250 PRINT TAB(14,0)"TIC-TAC-T
    OE"
1260 COLOUR 2
1270 PRINT '''"This is a simpl
    e game of Os and Xs."'
    "The computer plays with
     Os and you"''"play with
     Xs."''''"To play, enter
     the number of the square
     "''"you want."'
1280 PRINT '"If you try a
     square that has been
     taken"''"the computer
     will beep."
1290 PRINT TAB(10,29)"PRESS
     SPACE TO PLAY"
    :REPEAT UNTIL 32=GET
1300 ENDPROC
1310 DEF PROCinit
1320 DIM Z$(9)
1330 ENVELOPE 1,1,20,-20
    ,20,8,8,8,127,127,0
    ,0,127,127
1340 ENVELOPE 2,1,90,-10
    ,30,3,6,12,127,127,0
    ,0,127,127
1350 ENDPROC
1360 DEF FNtry(1$)
1370 RESTORE 1520
1380 found=FALSE
1390 LOCAL X,Y,Z
1400 FOR N%=1 TO 9
1410 READ X,Y,Z
1420 IF FNt(X,Y,Z)
    THEN found=TRUE
1430 NEXT
1440 =found
1450 DEF FNt(x,y,z)
1460 =(Z$(x)=1$ AND Z$(y)=1$
     AND Z$(z)=1$)
1470 DEF FNtest(X,Y,Z,L$)
1480 IF (Z$(X)="." AND Z$(Y)=L
    $ AND Z$(Z)=L$)
    THEN =TRUE
1490 IF (Z$(X)=L$ AND Z$(Y)=
    "."AND Z$(Z)=L$)
    THEN =TRUE
1500 IF (Z$(X)=L$ AND Z$(Y)=L$
     AND Z$(Z)=".")
    THEN =TRUE
1510 =FALSE
1520 DATA 1,2,3,4,5,6,7,8
    ,9,1,4,7,2,5,8,3,6,9
    ,1,5,9,3,5,7,0,0,0
1530 DATA 15,12,19,12,23
    ,12,15,16,19,16,23,16
    ,15,20,19,20,23,20
1540 DATA 1,9,9,1,3,7,7,3
    ,0,0
1550 DATA 1,9,2,8,3,7,4,6
    ,0,0
1560 DATA 2,4,1,2,3,6,8,6
    ,9,8,4,7,0,0,0
```

This listing is included in this month's cassette tape offer. See order form on Page 47

This maths workout is based on articles that originally appeared in The Micro User. Our thanks to our "big brother" magazine for permission to use it.

MIKE BIBBY'S

MATHS workout

Exercises for the Electron

WE have seen that we can code our numbers in ways other than our usual denary, or decimal, system.

We also looked last month at a way of coding known as the binary system, which uses the digits 0 to 1 to represent any number – unlike the denary system which uses the digits 0 to 9.

To distinguish the two systems, we decided to prefix binary numbers with the symbol "%".

The number "one hundred and sixty two" is encoded in each system as follows:

In denary,

162 i.e. 100+60+2

In binary,

```
128 64 32 16  8  4  2  1
 %  1  0  1  0  0  0  1  0
```

i.e. 128+32+2

Each column in the binary system, known as a "bit", contains either a one or a zero.

Although the binary representation of a number is rather cumbersome to write, this simple two-state system is easily represented by electrical circuits – which are either ***on*** or ***off.***

We saw that the computer handles bits in groups of eight at a time.

Such a group is called a byte. Thus a byte contains eight bits labelled, somewhat peversely, bits 0 to 7. *(See Figure I.)*

Bit 0, as you can see, is the "1" column.

As this is the smallest value bit we say that bit 0 is the least significant bit (LSB). Bit 7, the "128" column, is called the most significant bit (MSB).

The reason for using the numbers 0 to 7 to label the bits instead of the more logical 1 to 8 has to do with powers, a subject you almost certainly covered at school.

"2 to the power 2" is 2*2 = 4
"2 to the power 3" is 2*2*2 = 8
"2 to the power 4" is 2*2*2*2 = 16

and so on. "2 to the power 8" would be eight twos all multiplied together.

Notice as the powers of two increase – that is, as we multiply more twos together – the answers are doubling, just as our column or bit values do.

Also, 2 to the power of 2 is 4, the value of bit 2, while 2 to the power of 3 is 8, the value of bit 3.

It shouldn't come as any surprise to you to find that 2 to the power of 7 is 128, the value of bit 7.

You can verify this on the Electron by using the symbol "^" on the "↑" key which stands for "to the power of".

Try:

```
PRINT 2^4
PRINT 2^7
```

Be sure to try 2^1, which will show you why bit 1 has the value 2.

Also try 2^0. The answer may surprise you.

The fact is that any number to the power 0 is 1! Hence bit zero has the column value of one. Figure II illustrates this.

Look at this sum:

```
  % 1
+ % 1
-----
 %10
-----
```

If you think about it, that is correct, since the sum adds one and one, and the answer %10 is binary for two.

One way of relating this to our usual way of doing sums is to say that we carry when we get to two, instead of ten as we do in our normal, decimal, sums.

Another way to look at it is that we have to carry when we get to two because we aren't allowed to use the digit '2'.

If you remember, last month we had a rule forbidding two "coins" of the same value.

Try this sum:

```
    4 2 1
  %   1 1      ( 3 )
+ %   1 0      ( 2 )
-----------    -----
  % 1 0 1      ( 5 )
-----------    -----
```

Here we carry from the second column to the third.

Addition is not very hard at all – just make sure that you always "put 0 down and carry 1" when you get a two.

If you get a three then "carry one for two and put one down".

For example:

```
    8 4 2 1
  %   1 1 1      ( 7 )
+ %     1 1      ( 3 )
-------------    -----
  % 1 0 1 0      ( 10 )
-------------    ------
```

Subtraction is a little more complicated, and depends on whether you borrow or decompose!

The latter phrase doesn't

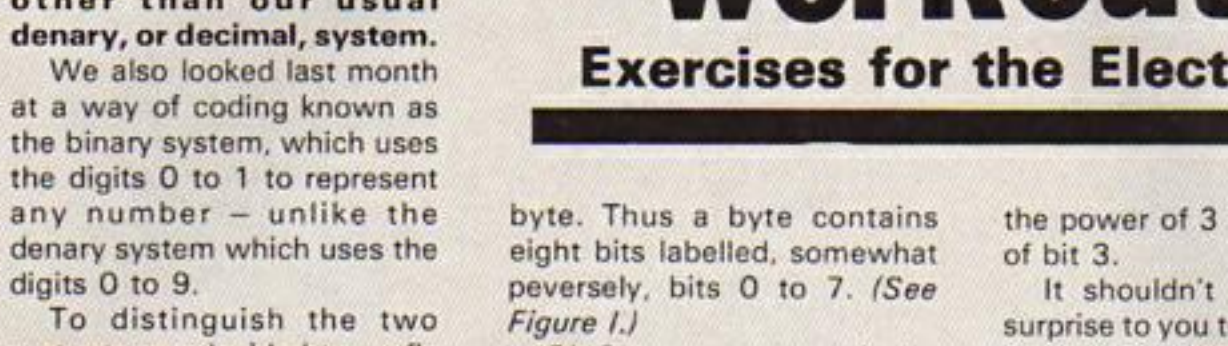


Figure I: The bit pattern for 141

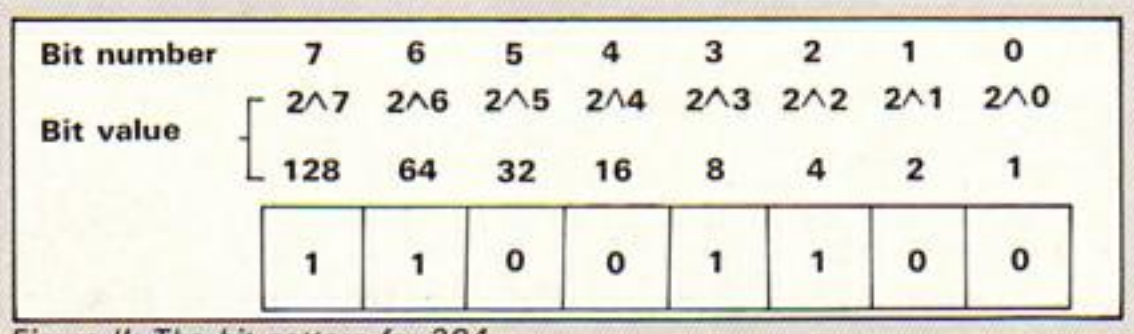
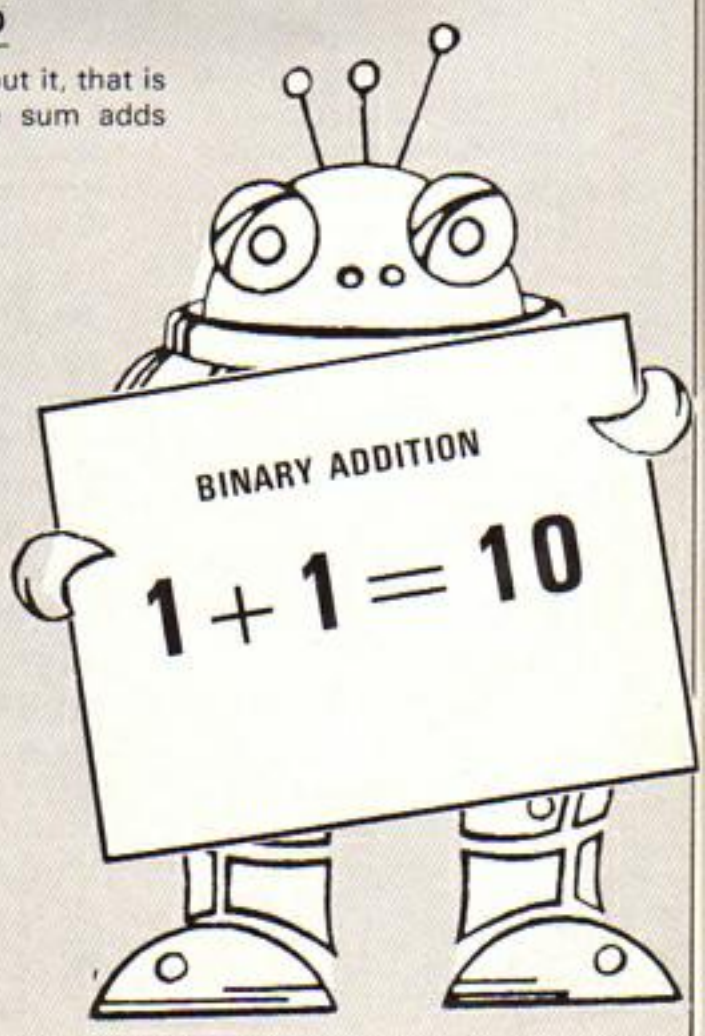


Figure II: The bit pattern for 204



describe the current economic climate, it's just that there are two schools of thought on the way subtraction should be taught – the borrowers and the decomposers.

Fortunately, we can ignore binary subtraction since we can manage without it – as does the microprocessor inside your machine.

If you want to do some binary subtraction it is straightforward enough provided that you remember that it is two you're borrowing or taking, not ten.

Figure III illustrates the process – without any attempt to explain it.

Before we leave the realm of simple sums, look what happens if we shift everything in a binary number over to the left, putting a zero into bit 0, which would be left vacant otherwise. For example:

```
    8 4 2 1
 %    1 0 1     which is 5
becomes
    8 4 2 1
 % 1 0 1 0      which is 10
```

```
 %  °1  '1°  'Ø          %  1  '1   'Ø          6
-%     1    1    OR    -%     '1°    1    In decimal -3
-------------          -------------          ---
 %     1    1          %     1    1             3
-------------          -------------          ---
Decomposition          Borrowing
```

Figure III: Binary subtraction

This shifting to the left doubles the number automatically.

This isn't too hard to visualise, because the value of each bit is transferred to the next higher bit, which is of course double in value – so the end result is that the whole number is doubled in value.

Similarly, we can do the binary equivalent of DIV 2 by shifting to the right. For example:

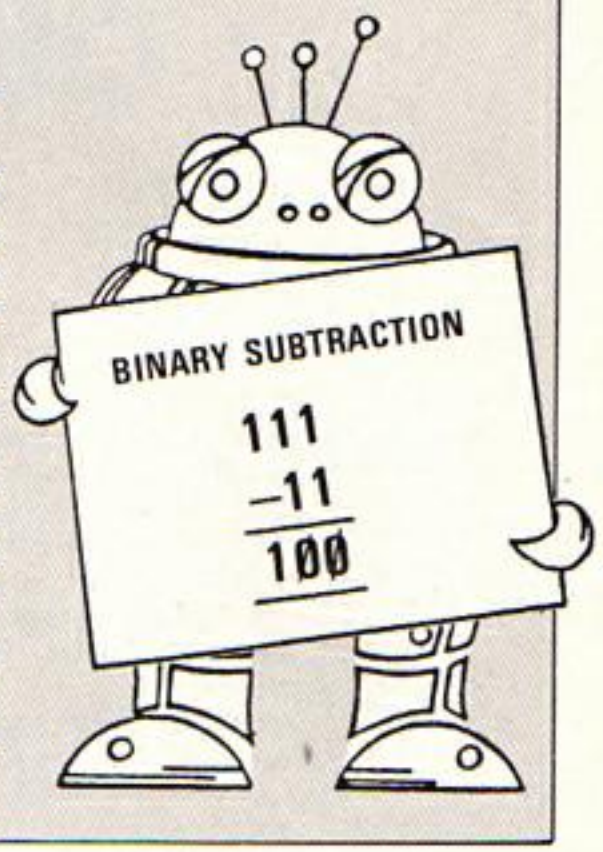
```
    8 4 2 1
 % 1 1 0 1     which is 13
becomes
    8 4 2 1
 %    1 1 0     which is 6
```

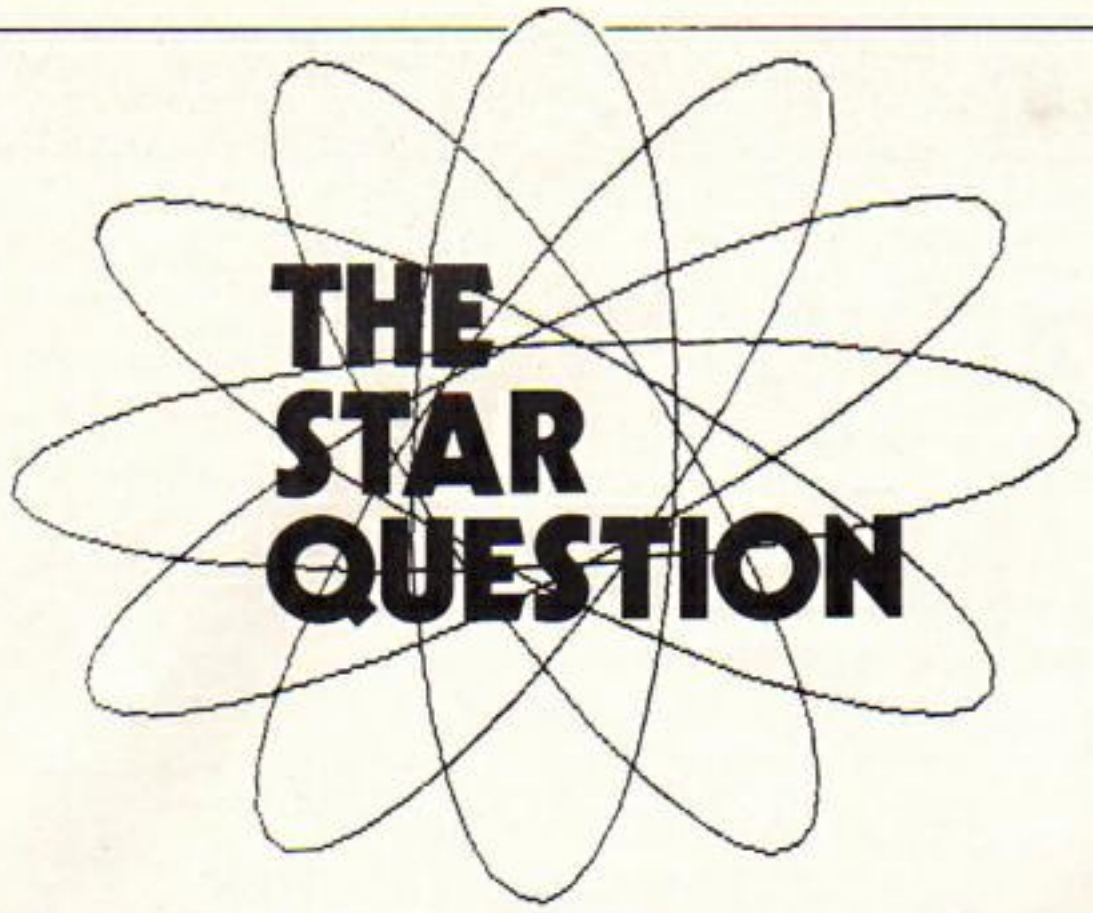
and, of course, 13DIV 2 gives you 6.

The DIV command, in case you aren't familiar, deals with integer division. That is, it does division but only tells you the "wholes" and ignores the remainders.

As each bit is moved to the right, it occupies a column exactly one half lower in value, thus the sum total of all the bits is one half lower, save for the original bit 0 which has disappeared altogether (hence the ignored remainder).

Well, that's enough binary for one month. Hexadecimal blooms in June!





THE STAR QUESTION

WHILE walking past the Micro User offices (known as the "bunker" to all and sundry), I was shocked to see the Editor actually doing some programming!

Not only that, but it worked!

Anyway, during the three or four hours he was out at lunch I nipped into his office, cleared away the empties and put a tape in the cassette and SAVEd it.

Obviously anything that BBC Micro owners have is fair game for us morally superior Electron users, so here it is. Can you figure out how he did it? I don't mean how it works, but how HE did it!

```
 10 REM SHELL
 20 REM BY MIKE BIBBY
 30 REM (C) ELECTRON USER
 40 REM WITH THANKS TO
 50 REM THE MICRO USER
 60 MODE 0
    :VDU 29,500;500;
 70 FACTOR=0.25
    :FSET=0.75
 80 FOR FSET=0 TO 2.5
    STEP 0.5
 90 FOR CIRCLE=0 TO 2*
    PI +0.2 STEP 0.2
100 XPOS=(COS (FSET)*
    COS (CIRCLE)-SIN (FSET)*
    SIN (CIRCLE)*FACTOR)*500
110 YPOS=(COS (CIRCLE)*
    SIN (FSET)+SIN (CIRCLE)*
    COS (FSET)*FACTOR)*500
120 IF CIRCLE=0
    THEN MOVE XPOS,YPOS
130 DRAW XPOS,YPOS
140 NEXT
    :NEXT
```

Quick Draw listing

From Page 29

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter is given on Page 4 of the February issue.

```
10 REM ELECTRON DRAUGHTSMAN
20 REM By Mike Cook
30 REM (C) ELECTRON USER
40 *KEY0 MODE 6|M *FX4
   ,0|M *FX12,0|M
50 REM PRESS FUNCTION KEY0
    TO REGAIN EDITING
60 M%=1
   :C%=3
70 MODE M%
80 TRIANGLE=FALSE
90 DAFT=FALSE
100 *FX11,0
110 *FX4,1
120 PR%=FALSE
130 DIM CLI% 40
140 CLS
150 PROC_INSTRUCTIONS
160 VDU 28,0,0,39,0
170 VDU 19,2,2,0,0,0
180 GCOL 3,C%
190 REPEAT
200 X%=50
   :Y%=50
210 REPEAT
220 A$=INKEY$ (0)
230 IF ASC (A$) > 134
   THEN A$=""
240 UNTIL A$<>""
250 FIRST%=TRUE
260 IF A$="W"
   THEN CLG
270 IF A$="S"
   THEN PROC_FILE
280 IF A$="G"
   THEN PROC_GET
290 IF A$="P"
   THEN PROC_POLY
300 IF A$="R"
   THEN PROC_REC
310 IF A$="C"
   THEN PROC_COLCHANGE
320 IF A$="L"
   THEN PRINT "LINE";
   :PROC_LINE
330 IF A$="T"
   THEN PROC_TRIANGLE
340 IF A$<>"C"
   THEN PRINT
350 UNTIL DAFT
360 DEF PROC_FILE
370 PRINT
380 INPUT "FILE NAME FOR
    SAVED SCREEN",F$
390 IF LEN (F$) < 1
   THEN ENDPROC
400 $CLI%="SAVE "+F$+" 3000
    8000"
410 X%= CLI% MOD 256
   :Y%=CLI% DIV 256
420 CALL &FFF7
430 ENDPROC
440 DEF PROC_GET
445 *D.
450 PRINT
460 INPUT "FILE NAME OF SCREE
   N",F$
470 IF LEN (F$) < 1
   THEN ENDPROC
480 $CLI%="LOAD "+F$
490 X%= CLI% MOD 256
   :Y%=CLI% DIV 256
500 CALL &FFF7
510 ENDPROC
520 DEF PROC_COLCHANGE
530 C%=(C%+1) AND 3
540 IF C%=0
   THEN C%=C%+1
550 COLOUR C%
   :GCOL 3,C%
560 PRINT "NEW COLOUR ";
570 ENDPROC
580 DEF PROC_BAND1
590 REPEAT
600 PROC_FOLLOW
610 A$=INKEY$ (0)
620 IF ASC (A$) > 134
   THEN A$=""
630 UNTIL A$<>""
640 OX%=X%
   :OY%=Y%
650 XT%=X%
   :YT%=Y%
660 ENDPROC
670 DEF PROC_BAND2
680 REPEAT
690 PROC_FOLLOW
700 MOVE OX%,OY%
710 PLOT 13,XT%,YT%
720 MOVE OX%,OY%
730 PLOT 13,X%,Y%
740 XT%=X%
   :YT%=Y%
750 A$=INKEY$ (0)
760 IF ASC (A$) > 134
   THEN A$=""
770 UNTIL A$<>""
780 MOVE OX%,OY%
790 PLOT 13,X%,Y%
800 ENDPROC
810 DEF PROC_LINE
820 REPEAT
830 IF A$<>"J"
   THEN PROC_BAND1
840 PROC_BAND2
850 GCOL 0,C%
860 MOVE OX%,OY%
870 DRAW X%,Y%
880 GCOL 3,C%
890 T1X%=OX%
   :T1Y%=OY%
900 OX%=X%
   :OY%=Y%
910 UNTIL A$=CHR$ (13)
   OR TRIANGLE=TRUE
920 ENDPROC
930 DEF PROC_TRIANGLE
940 PRINT "TRIANGLE";
950 REPEAT
960 TRIANGLE=TRUE
970 PROC_LINE
980 TRIANGLE=FALSE
990 REPEAT
1000 T2X%=X%
   :T2Y%=Y%
1010 PROC_TRIBAND(X%,Y%)
1020 REPEAT
1030 PROC_FOLLOW
1040 PROC_TRIBAND(OX%,OY%)
1050 OX%=X%
   :OY%=Y%
1060 PROC_TRIBAND(X%,Y%)
1070 A$=INKEY$ (0)
1080 IF ASC (A$) > 134
   THEN A$=""
1090 UNTIL A$<>""
1100 PROC_TRIBAND(X%,Y%)
1110 GCOL 0,C%
1120 PROC_TRIBAND(X%,Y%)
1130 GCOL 3,C%
1140 IF A$<>CHR$ (13)
   THEN PLOT 69,X%,Y%
1150 T1X%=T2X%
   :T1Y%=T2Y%
1160 UNTIL A$<>"J"
1170 UNTIL A$=CHR$ (13)
1180 ENDPROC
1190 DEF PROC_TRIBAND(X%
   ,Y%)
1200 MOVE T1X%,T1Y%
1210 PLOT 13,X%,Y%
1220 MOVE T2X%,T2Y%
1230 PLOT 13,X%,Y%
1240 ENDPROC
1250 DEF PROC_REC
1260 PRINT "RECTANGLE";
1270 REPEAT
1280 PROC_BAND1
1290 REPEAT
1300 PROC_RECBAND
1310 REPEAT
1320 PROC_FOLLOW
1330 PROC_RECBAND
1340 OX%=X%
   :OY%=Y%
1350 PROC_RECBAND
1360 A$=INKEY$ (0)
1370 IF ASC (A$) > 134
   THEN A$=""
1380 UNTIL A$<>""
1390 PROC_RECBAND
1400 GCOL 0,C%
1410 PROC_RECBAND
1420 GCOL 3,C%
1430 IF A$<> CHR$ (13)
   THEN PLOT 69,X%,Y%
1440 XT%=X%
   :YT%=Y%
1450 UNTIL A$<>"J"
1460 UNTIL A$=CHR$ (13)
1470 ENDPROC
1480 DEF PROC_RECBAND
1490 MOVE XT%,YT%
1500 PLOT 13,XT%,OY%
1510 PLOT 13,OX%,OY%
1520 PLOT 13,OX%,YT%
1530 PLOT 13,XT%,YT%
1540 ENDPROC
1550 DEF PROC_POLY
1555 REPEAT
1560 INPUT "POLYGON NUMBER
    OF SIDES",N%
1565 UNTIL N%>0
1570 PRINT N%;" SIDED POLYGON";
1580 REPEAT
1590 PROC_BAND1
1600 REPEAT
1610 PROC_BAND2
1620 PROC_DPOLY(X%,Y%,OX%
   ,OY%,N%)
1630 IF A$="J"
   THEN MOVE OX%,OY%
   :PLOT 13,X%,Y%
1640 UNTIL A$<>"J"
1650 UNTIL A$=CHR$ (13)
1660 PLOT 69,X%,Y%
1670 ENDPROC
1680 DEF PROC_DPOLY(X,Y,XT%
   ,YT%,N%)
1690 LOCAL C1,S1,P,R,A%,X1
   ,Y1
1700 GCOL 0,C%
1710 P=2*PI /N%
1720 N%=N%+1
1730 C1=COS (P)
1740 S1=SIN (P)
```

→

Quick Draw listing

From Page 57

```
1750 MOVE X,Y
1760 FOR A%=1 TO N%-1
1770 X1=XT%+(X-XT%)*C1-(Y-YT%)
     *S1
1780 Y1=YT%+(X-XT%)*S1+(Y-YT%)
     *C1
1790 X=X1
     :Y=Y1
1800 DRAW X,Y
1810 NEXT
1820 GCOL 3,C%
1830 PLOT 69,X%,Y%
1840 ENDPROC
1850 DEF PROC_FOLLOW
1860 IF FIRST%
     THEN FIRST%=FALSE
     ELSE PLOT 69,X%,Y%
1870 DEF PROC_CURS
1880 IF INKEY (-122)
     THEN X%=X%+SPEED%
     :PR%=TRUE
1890 IF INKEY (-26)
     THEN X%=X%-SPEED%
     :PR%=TRUE
1900 IF INKEY (-58)
     THEN Y%=Y%+SPEED%
     :PR%=TRUE
1910 IF INKEY (-42)
     THEN Y%=Y%-SPEED%
     :PR%=TRUE
1920 IF X%>1279 OR X%<0
     OR Y%>1023 OR Y%<0
     THEN PRINT
     :PRINT "X = ";X%;" Y
      = ";Y%;
1930 IF PR%
     THEN SPEED%=SPEED%+2

     ELSE SPEED%=1
1940 IF SPEED%> 30
     THEN SPEED%=30
1950 IF NOT (PR%)
     THEN 1970
1960 *FX15,1
1970 PR%=FALSE
1980 PLOT 69,X%,Y%
1990 ENDPROC
2000 DEF PROC_INSTRUCTIONS
2010 PRINT
2020 PRINT SPC (9);"ELECTRON
     DRAUGHTSMAN"
2030 PRINT SPC (13);"By Mike
     Cook"
2040 PRINT
2050 PRINT "First select a
     mode by typing a letter:
     -"
2060 PRINT
2070 PRINT "L - Draw a LINE"
2080 PRINT "T - Draw a TRIANGL
     E"
2090 PRINT "R - Draw a RECTANG
     LE"
2100 PRINT "P - Draw a POLYGON
     or CIRCLE"
2110 PRINT
2120 PRINT "Then move the
     dot with the cursor keys."
2130 PRINT
2140 PRINT "Press RETURN at
     the end of each stage"
2150 PRINT "or to stay in
     the mode press SPACE."
2160 PRINT "Alternatively
     pressing J as the last
     key";
2170 PRINT "will Join up the
     next shape."
2180 PRINT
2190 PRINT
2200 PRINT "Other commands
     are:-"
2210 PRINT
2220 PRINT "C - To change
     the COLOUR"
2230 PRINT "W - To Wipe the
     screen clean"
2240 PRINT "S - To SAVE the
     screen as a file"
2250 PRINT "G - To GET a scree
     n previously saved"
2260 PRINT
2270 PRINT "Press any key
     to begin."
2280 A$=GET$
2290 CLS
2300 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 47

Maths Hike listing

From Page 37

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
     THEN PRINT TAB(20
     ,7)"   "
370 UNTIL limit%>1
380 INPUT TAB(3,10) "What
     level of difficulty?(1
     -9)"TAB(20,12) difficul
     ty
390 IF difficulty <1
     OR difficulty >9
     THEN PRINT TAB(20
     ,12)"   "
     :GOTO 380
400 INPUT TAB(3,15)"How
     many calculations
     do you want?"TAB(20
     ,17)turns
410 FOR delay= 1 TO 1000/di
     fficulty
     :NEXT delay
420 CLS
430 ENDPROC
440 DEF PROChike
450 total=RND(limit%)
460 PRINT TAB(12,15) total
470 SOUND &11,-15,100
     ,2
480 sum$=STR$ (total)

490 FOR delay= 1 TO 2000/di
     fficulty
     :NEXT delay

500 FOR goes=1 TO turns
510 CLS
520 sum$=STR$ (total)
530 chance=RND(4)
540 IF chance=1
     THEN operator$="+"
550 IF chance=2
     THEN operator$="-"
560 IF chance=3
     THEN operator$="*"
570 IF chance=4
     THEN operator$="/"
580 number$=STR$ (RND(limit
     %))
590 total=EVAL (sum$+operat
     or$+ number$)
600 PRINT TAB(20,15) operat
     or$+ number$
610 SOUND &11,-15,100
     ,2
620 FOR delay= 1 TO 2000/di
     fficulty
     :NEXT delay
630 sum$=STR$ (total)
640 NEXT
650 ENDPROC
660 DEF PROCanswer
670 CLS
680 INPUT TAB(5,10) "What's
     the answer?"TAB(19
     ,13) answer
690 IF answer=EVAL (sum$)

     THEN PRINT TAB(5,17)
      "Correct."
     :ENVELOPE 2,2,6,0
     ,0,255,0,0,126,0,0
     ,-126,126,126
     :SOUND 1,2,4,50
700 IF answer<>EVAL (sum$)
     THEN PRINT TAB(5,15)
      "Wrong. The answer
      was ";EVAL (sum$)
     :SOUND 0,-15,2,10
710 PRINT TAB(5,22) "Press
     any key for another
     go"
720 PRINT TAB(5,24) "Press
      ESCAPE to change
     levels"
730 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 47

Chaser listing

From Page 27

```
    THEN PROCend("TWO")
680 X1%=X1%+X3%
690 Y1%=Y1%+Y3%
700 ENDPROC
710 :
720 :
730 DEF PROChelp
740 Y1%=Y1%-Y3%
750 X1%=X1%-X3%
760 X3%=0
770 Y3%=0
780 IF POINT(X1%*32+16
    ,Y1%*32+16)=0
    THEN C1%=225
    :X3%=0
    :Y3%=1
790 IF POINT(X1%*32+48
    ,Y1%*32-16)=0
    THEN C1%=227
    :X3%=1
    :Y3%=0
800 IF POINT(X1%*32+16
    ,Y1%*32-48)=0
    THEN C1%=226
    :X3%=0
    :Y3%=-1
810 IF POINT(X1%*32-16
    ,Y1%*32-16)=0
    THEN C1%=228
    :X3%=-1
    :Y3%=0
820 ENDPROC
830 :
840 :
850 DEF PROCc2
860 PROChe2
870 IF X4%=0 AND Y4%=0
    THEN PROCend("ONE")
880 X2%=X2%+X4%
890 Y2%=Y2%+Y4%
900 ENDPROC
910 :
920 :
930 DEF PROChe2
940 X2%=X2%-X4%
950 Y2%=Y2%-Y4%
960 X4%=0
970 Y4%=0
980 IF POINT(X2%*32+16
    ,Y2%*32+16)=0
    THEN C2%=225
    :X4%=0
    :Y4%=1
990 IF POINT(X2%*32+48
    ,Y2%*32-16)=0
    THEN C2%=227
    :X4%=1
    :Y4%=0
1000 IF POINT(X2%*32+16
     ,Y2%*32-48)=0
     THEN C2%=226
     :X4%=0
     :Y4%=-1
1010 IF POINT(X2%*32-16
     ,Y2%*32-16)=0
     THEN C2%=228
     :X4%=-1
     :Y4%=0
1020 ENDPROC
1030 :
1040 :
1050 DEF PROCend(W$)
1060 IF W$="ONE"
     THEN X1%=X1%+X3%
     :Y1%=Y1%+Y3%
     :PROChelp
1070 IF W$="ONE" AND X3%=0
      AND Y3%=0
     THEN W$="DRAW"
1080 IF W$="TWO"
     THEN X2%=X2%+X4%
     :Y2%=Y2%+Y4%
     :PROChe2
1090 IF W$="TWO" AND X4%=0
      AND Y4%=0
     THEN W$="DRAW"
1100 FOR W=1TO 2500
1110 NEXT
1120 VDU 4
1130 IF W$="DRAW"
     THEN PRINT TAB(10
     ,10);"A draw"
     :GOTO 1180
1140 PRINT TAB(10,10);"Playe
     r ";W$;" won."
1150 PROCs("bBBABcCbBaAAGAbB
     gGbBBABcDeEdDcAgGg"
     ,2)
1160 IF W$="ONE"
     THEN G%=G%+1
1170 IF W$="TWO"
     THEN H%=H%+1
1180 FOR W=1TO 2500
1190 NEXT
1200 *FX15,0
1210 PROCscores
1220 PROCfinish
1230 ENDPROC
1240 :
1250 :
1260 DEF PROCIntro
1270 PRINT TAB(12,4);"******
     **********"
1280 PRINT TAB(12,5);"***
      CHASER  ***"
1290 PRINT TAB(12,6);"******
     **********"
1300 PRINT TAB(3,10);"Design
     ed and written by :"
1310 PRINT TAB(9,13);"******
     ****************"
1320 PRINT TAB(9,14);"***
      James Mcpherson ***"
1330 PRINT TAB(9,15);"***
                     ***"
1340 PRINT TAB(9,16);"**
     & Peter Mitchell.
     **"
1350 PRINT TAB(9,17);"******
     ****************"
1360 PRINT TAB(14,22);"PRESS
     ANY KEY"
1370 A=GET
1380 CLS
1390 PRINT TAB(13,3);"**
     CHASER **"
1400 PRINT 'TAB(3);"This
     is a game for two
     players."
1410 PRINT 'TAB(3);"Player
     one starts with the
     arrow on"
1420 PRINT ;TAB(3);"the
     left while player
     two starts"
1430 PRINT TAB(3);"with
     the arrow on the right
     ."
1440 PRINT 'TAB(3);"During
     the game if either
     of the"
1450 PRINT TAB(3);"players
     tries to cross a path
     left"
1460 PRINT TAB(3);"their
     opponent or themselves
     it"
1470 PRINT TAB(3);"cannot
     be done."
1475 PRINT 'TAB(3) "The
     loser is the first
     one to run"
1476 PRINT TAB(3) " out
     of room."
1480 PRINT '''TAB(10);"PRESS
     ANY KEY"
1490 A$=GET$
1500 CLS
1510 PRINT 'TAB(12,5);"**
     CONTROLS **"
1520 PRINT '''TAB(8);"Player
     1 :";TAB(21);"Player
     2 :"
1530 PRINT '"left"
1540 PRINT "right"
1550 PRINT "up"
1560 PRINT "down"
1570 PRINT TAB(11,11);"
     Z ";TAB(24,11);" <
     "
1580 PRINT TAB(11,12);"
     X ";TAB(24,12);" >
     "
1590 PRINT TAB(11,13);"
     Q ";TAB(24,13);" *
     "
1600 PRINT TAB(11,14);"
     A ";TAB(24,14);" ?
     "
1610 PRINT TAB(8,20);"Press
     any key to begin."
1620 A$=GET$
1630 PROCs("CCDECEDGCCDECCB6
     CCDEFEDCBGABCCC".4)
1640 ENDPROC
1650 :
1660 :
1670 DEF PROCfinish
1680 FOR W%=1 TO 500
1690 A=INKEY (0)
1700 NEXT
1710 A=GET
1720 CLS
1730 CLEAR
     :GOTO 90
1740 END
1750 ENDPROC
1760 :
1770 :
1780 DEF PROCscores
1790 VDU 4
1800 VDU 19,0,6,0;0
1810 CLS
1820 COLOUR 130
     :COLOUR 1
1830 PRINT TAB(16,2);"CHASER
     "
1840 PRINT TAB(8,7);"*******
     ****************"
1850 PRINT TAB(8,8);"* Playe
     r 1.....";G%;
     " *"
1860 PRINT TAB(8,9);"*******
     ****************"
1870 PRINT TAB(8,11);"******
     ****************"
1880 PRINT TAB(8,12);"*
      Player 2.....";H
     %;" *"
1890 PRINT TAB(8,13);"******
     ****************"
1900 PRINT TAB(7,23);"Press
      any key to continue";
1910 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 47

Coder listing

From Page 22

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
160 DEF PROCINIT
170 READ PASSWORD$
180 DIM PLACE(6)
190 ENDPROC
200 DEF PROCIDENTIFY
210 ATTEMPTS=0
220 PRINT '" ELECTRON EDDIES
    SECRET CODING MACHINE"'
    " **********************
    **************"
230 VDU 7
    :PRINT '''"Type in the
    password"
240 PRINT TAB((40-LEN (PASSWO
    RD$))/2,10)STRING$(
    LEN (PASSWORD$),"-")
250 PRINT TAB((40-LEN (PASSWO
    RD$))/2,10);
260 IDENTITY$=""
    :*FX15,1
270 FOR I%=1TO LEN (PASSWORD$)
280 IDENTITY$=IDENTITY$+GET$
290 PRINT SPC (1);
300 NEXT
310 IF PASSWORD$=IDENTITY$
    ENDPROC
320 ATTEMPTS=ATTEMPTS+1
330 IF ATTEMPTS<3
    THEN VDU 7
    :CLS
    :PRINT "WRONG PASSWORD!..
    ....TYPE IT AGAIN!"
    :GOTO 240
340 VDU 7
    :CLS
350 PRINT '''''"ILLEGAL OPERAT
    ION....."''"SORRY, YOU
    CAN'T USE THE PROGRAM"
    :GOTO 350
360 DEF PROCNUMBER
370 VDU 7
    :CLS
    :INPUT ''''"Enter your
    code number and press
    RETURN"''"    "N$
380 IF LEN (N$)<>6 PROCINVALI
    D
    :GOTO 370
390 FOR I%=1TO 5
    :IF INSTR(N$,STR$ (I%))=0
    I%=6
    :PROCINVALID
    :GOTO 370
400 NEXT
410 IF VAL (RIGHT$(N$,1))<0
    OR VAL (RIGHT$(N$,1))>4
    PROCINVALID
    :GOTO 370
420 ENDPROC
430 DEF PROCINVALID
440 VDU 7
    :CLS
    :PRINT ''"YOUR CODE NUMBE
    R WAS INVALID!"
450 PRINT ''"It must consist
    of six numbers!"''
    "The first five must
    be the numbers from
    1 to 5 rearranged in
    some order."
460 PRINT ''"The last must
    be a number from 0 to
    4"
470 PRINT ''"Here are some
    examples that will work:
    -"''"142354    245312
        123540    254313"
480 PRINT ''"See Electron
    User for more details."
490 PRINT ''"Press any key,th
    en enter your code again"
    :A=GET
    :ENDPROC
500 DEF PROCTYPEIN
510 VDU 7
    :CLS
    :PRINT ''"Type in your
    message"''"The maximum
    length is 250 characters
    "''"The present length
    is"''"Press RETURN to
    end the message"
520 PART$=""
    :MESSAGE$=""
530 REPEAT
540 PRINT TAB(LEN (MESSAGE$)
    MOD 40,10+LEN (MESSAGE$)
    DIV 40)PART$
550 IF ASC (PART$)=127
    THEN MESSAGE$=LEFT$(MESSA
    GE$,LEN (MESSAGE$)-1)
    ELSE MESSAGE$=MESSAGE$+PA
    RT$
560 PRINT TAB(22,6)STR$ (
    LEN (MESSAGE$))
570 PART$=GET$
580 IF LEN (MESSAGE$)>=240
    PRINT TAB(0,20)"MESSAGE
    NEARLY MAXIMUM LENGTH"
    :VDU 7
590 FOR I%=1TO 6
600 PLACE(I%)=VAL (MID$(N$
    ,I%,1))
610 NEXT
620 UNTIL PART$=CHR$ (13)
    OR LEN (MESSAGE$)=250
630 ENDPROC
640 DEF PROCCODE
650 PROCTYPEIN
660 VDU 7
    :CLS
    :PRINT TAB(5,10)"MESSAGE
    IS BEING CODED"
670 C%=0
    :CODE$=""
680 REPEAT
690 FOR I%=1TO 5
700 CODE$=CODE$+MID$(MESSAGE$
    ,C%+PLACE(I%),1)
710 NEXT
720 C%=C%+5
730 UNTIL C%>LEN (MESSAGE$)
740 FINALCODE$=""
750 FOR I%=1TO LEN (MESSAGE$)
760 FINALCODE$=FINALCODE$+
    CHR$ (ASC (MID$(CODE$
    ,I%,1))+PLACE(6))
770 NEXT
780 CLS
    :VDU 7
    :PRINT "Your message:-"'M
    ESSAGE$''"Has been coded
    to:-"'FINALCODE$''"Do
    you want to save it
    on tape (Y OR N)?"'
790 ANSWER$=GET$
    :IF ANSWER$="N"
    THEN 840
    ELSE IF ANSWER$<>"Y"
    THEN 790
800 *OPT1,1
810 X%=OPENOUT ("CODE")
820 PRINT #X%,FINALCODE$
830 CLOSE #X%
840 ENDPROC
850 DEF PROCDECODE
860 VDU 7
    :CLS
    :PRINT '"Do you wish
    to enter the coded messa
    ge"''"from the keyboard,
    or tape (K or T)"
870 ANSWER$=GET$
    :IF ANSWER$="K"PROCTYPEIN
    :GOTO 920
    ELSE IF ANSWER$<>"T"
    THEN 870
880 *OPT1,1
890 X%=OPENIN ("CODE")
900 INPUT #X%,MESSAGE$
910 CLOSE #X%
920 FINALCODE$=MESSAGE$
930 FOR I%=1TO 5
940 PLACE(I%)=INSTR(N$,
    STR$ (I%))
950 NEXT
960 PLACE(6)=VAL (RIGHT$(N$
    ,1))
970 VDU 7
    :CLS
    :PRINT TAB(5,10)"MESSAGE
    IS BEING DECODED"
980 CODE$=""
990 FOR I%=1TO LEN (FINALCODE
    $)
1000 CODE$=CODE$+CHR$ (
    ASC (MID$(FINALCODE$
    ,I%,1))-PLACE(6))
1010 NEXT
1020 C%=0
    :MESSAGE$=""
1030 REPEAT
1040 FOR I%=1TO 5
1050 MESSAGE$=MESSAGE$+
    MID$(CODE$,C%+PLACE(I%)
    ,1)
1060 NEXT
1070 C%=C%+5
1080 UNTIL C%>LEN (CODE$)
1090 CLS
    :VDU 7
    :PRINT '"Your coded messa
    ge:-"'FINALCODE$''"Decode
    s to:-"'MESSAGE$
1100 ENDPROC
1110 DATA ELECTRON
```

This listing is included in this month's cassette tape offer. See order form on Page 47

Micro Messages

Software solution to *TV255 poser

*IN REPLY to Mr Bobut's letter in the March 1984 Electron User lamenting the lack of a *TV255 on the Electron. I suggest that he tries:*

```
VDU 28,0,24,39,1
```

which, in Mode 6, will get rid of the top line of text and so make his listings more readable.

This puts the text in a window which is the same as the screen except for the top line.

```
*KEY0"VDU28,0,24,
39,1|M
```

puts this utility on the 0 function key. – **K. Goodacre, Sheffield.**

● Many thanks for this software solution.

Hardware alternative

*REGARDING the lack of Electron *TV commands, Acorn recommend reducing the height of the picture on the TV.*

On newer TVs this adjustment has to be carried out inside the set, but on older sets you might have to take it to a TV shop. – **I. Gardner, Sandwich, Kent.**

● Thanks for giving us the hardware alternative to the problem.

Simple saving remedy

DO any readers have problems saving and loading programs?

If, like me, you get the dreaded 'Data?, Block?, Rewind tape' messages, I think there may be a simple remedy.

As recommended I connected my Electron to the Mic input on my tape recorder. Due to its sensitivity this distorted the recorded signal, so the computer couldn't always read it properly when loading.

So I tried connecting it instead to the tape recorder's other input socket, usually marked Aux. or Line Input.

This worked superbly, and I now save and load programs with ease.

An alternative would be to put a resistor in series with the recorder's Mic input to cut down the signal strength. Something between 1 to 5k ohm should work.

Resistors are only 4p each, and much cheaper than a new recorder.

Also a ¼ watt resistor should fit neatly inside a solder tag jack plug. – **L.J. Goodridge, Leeds.**

● Many thanks for your tip. Has anyone else any helpful advice about cassette difficulties that they'd like to share?

Adding more colour

I'D like to make a comment about the DRAW program in Notebook (March 1984 Electron User).

During each pass through the nested loop lines 50, 60 and 70 determine the three colours from which line 140 chooses.

This restricts the number of colours to three and, because of the random nature of lines 50, 60 and 70, some of these colours can be the same.

This can be avoided by putting in the following lines:

```
50 VDU 19,1,
   RND(3),0,0,0
60 VDU 19,2,
   RND(2)+3,0,0,0
70 VDU 19,3,
   RND(2)+5,0,0,0
```

This avoids the duplication of colours. – **A. Farmer, Warrington.**

● Many thanks for your new lines. We like to hear of improvements to our listings.

Cash-in with the Count

IN the January 1984 Electron User there is an article called "Going Quackers" in which two head shapes are shown, VDU 227 and VDU 228.

On running the program I saw that VDU 228 wasn't used, so I made my first attempt at programming.

I inserted three extra lines as follows:

```
361 VDU 17,2,228,
    10,8,17,0,
    231,233,10,8,
    234,234,8,8,
    8,232,232
362 PROCDelay
363 VDU 9,127,
    127,127,11,
    9,9,9,127,
    127,127,11,
    ,9,9,127,
    127,127,127
```

Now the duck stops and turns its head. I thought you might be interested. – **Graeme J. Cole, Leyton, London.**

● Nice one Graeme. If that's your first attempt at programming we're looking forward to the others. Incidentally, has anyone got a better quacking sound?

Stopping the duck!

THE Count program in the February 1984 issue of Electron User can become a very neat cash account with one or two modifications. The altered listing is as follows:

```
10 REM CASH
   ACCOUNT
20 REM BY
   W.J.DAVIES
30 PRINTTAB(16);
   "CASH ACCOUNT"
40 PRINTTAB(16);
   "------------"
50 PRINT
60 total=0
70 REPEAT
80 PRINT
90 INPUT"Amount?"
   TAB(16)
   "-"number
100 total=
    total+number
110 UNTIL
    number=0
120 PRINT
130 PRINTTAB(7);
    "Balance";
    "..-";total
```

Just run the program and type in the amounts required. For cash paid out use the minus sign before the figures.

After you've entered all your receipts and payments, key in 0, press Return and you have your balance in hand. – **W.J. Davies, Sidcup, Kent.**

● Many thanks for the program Mr Davies. It was a nice idea to send us the listing on some double-entry paper!

Use GOTOs properly

I HAVE been reading with some amusement the many arguments about structured programming. Somebody should explain to everyone what it really is!

A structured program can have as many GOTOs and GOSUBs as you want – as long as they are used properly.

I used to have a Jupiter Ace and spent a year programming in Forth. I can only write structured programs. – **R.A. Waddilove, Widnes.**

● This is an argument that seems to have spilled over from the pages of *The Micro User.*

Some people love structured, others hate it. Still others try to be structured but slip into the occasional GOTO.

What do our readers think? Do you care? And what micros (if any) did you have before your Electron and how did they compare?

Is this a record?

I SCORED 106,300 recently on the Micro Power game Killer Gorilla.

I was wondering if this was the highest

→

Micro Messages

From Page 61

score so far, after reading that the hi-score was 68,300. – **David Moffat, Methil, Fife.**

● Well done David, it's nice to hear of someone so skilled. No one at *Electron User* will admit their scores, and we've certainly not come across any higher one. No doubt we shall hear.

Flowers after just 7 weeks

I AM sending you this short program, hoping it will be of some interest.

It produces flowers of red, yellow and cyan on a green background, clearing when 300 have "grown".

I'm sure you will not find it perfect, as it's my first effort – we have only been Electron owners for seven weeks.

If nothing else, the character definitions for the flower (see lines 80-110) may be of some use. – **Mrs June Griffin, Royston, Herts.**

```
10 REM FLOWERS
20 REM  BY
   J.K.GRIFFIN

30 MODE 1
40 VDU 23,1,0;
   0;0;0;
50 VDU 19,0,2,
   0,0,0
60 VDU 19,3,6,
   0,0,0
70 COLOUR 128
80 VDU
23,225,249,
   124,46,31,63,
   62,30,12
90 VDU 23,226,62,
   124,232,240,
   248,248,240,
   96
100 VDU 23,227,3,
    7,15,15,55,
    127,254,253
110 VDU 23,228,
    128,192,224,
    224,216,252,
254,126
120 REPEAT
130 count=0
140 REPEAT
150 A=RND(3)
160 COLOUR A
170 count=
    count+1
180 X=RND(36)+1
    :Y=RND(30)
190 PRINT
    TAB(X,Y)
    CHR$(225);
    CHR$(226)
200 PRINT
    TAB(X,Y-1)
    CHR$(227);
    CHR$(228)
210 UNTIL
    count=30
220 FOR pause=1
    TO 1000:NEXT
230 CLS
240 UNTIL FALSE
```

Keep clear of "Bad mode"

WHEN I program on my Electron in "normal" Basic I seem to be able to enter and leave an 80 column mode without difficulty.

If, however, I use the procedure method of programming, I get "Bad mode" coming up whether I try to enter the 80 column mode from within the procedures or from without.

Can you tell me what I am likely to be doing wrong? – **J.M. Layton, Wellingborough.**

● The short answer is that you can't change your mode in a procedure.

The Electron uses part of its memory as a sort of electronic scrap pad. Here it keeps track of things like the variables used.

When you change mode in a procedure, the use of memory is changed and the scrap pad can be overwritten – with the results you have seen!

Having said that, we have little doubt that we'll be inundated with letters telling us how to do it!

● Thanks for the program – not bad after only seven weeks!

Going round in circles . . .

IN the March 1984 Micro Messages Hasan Bobut wanted to know how to draw circles. I've written a short program that will do this:

```
10 MODE 4
20 FOR A=0 TO
   2*PI
   STEP 0.01
30 PLOT 69,649+
   440*SIN(A),
   512+400 *
   COS(A)
40 NEXT
```

– **Brian Lord, Erith, Kent.**

● Thanks for the program, Hasan's letter certainly generated a lot of interest.

Bad program made good

BY mistake we have recorded over the end of a very long program and we are now getting a "Bad program" error.

Is there any way that we can copy the listing from tape into the computer so that we can re-enter the program lines that have been deleted?

We tried to use "File" but this didn't work. – **Sarah and Rachel Boxall, Stansted, Essex.**

● Much as it pains us to refer to it, you'll find the solution to your problem in Frank Dart's article on page 113 in the March 1984 issue of *The Micro User*.

This brings up another point. What do our readers think about our reprinting some of the more relevant articles that have appeared in *Micro User* in our far superior publication?

So far we've stuck to material that was published in *The Micro User* before *Electron User* existed. Should we change this policy? Over to you.

DO you like us or do you hate us? Are our games too hard or too easy? And what about the articles?

Write to us at Micro Messages and tell us. We can take it!

Remember, that these are the pages that you write yourselves. So tear yourself away from your Electron keyboard and drop us a line.

The address is:

Micro Messages
Electron User
Europa House
68 Chester Road
Hazel Grove
Stockport
SK7 5NY.

SPECIAL OFFER!
Deduct £1 per cassette when ordering 2 or more.

TOP QUALITY SOFTWARE FOR THE ACORN ELECTRON

ACORN ELECTRON

CENTIBUG £7.95
The centibug descends from the top of the screen weaving intimidatingly between the mushrooms. Your objective is to shoot all the segments of the centibug before it reaches the bottom of the screen.
Features include: spiders, snails, flies, 6 skill levels, hi-score, rankings, and increasing difficulty.

ALIEN DROPOUT £7.95
A novel and unusual program. Arcade-action with this exciting multi-stage shooting game. The objective of the game is to shoot the aliens out of their "boxes" before the "boxes" fill up. Once full, the aliens fly down relentlessly, exploding as they hit the ground. The game features include: 6 skill levels, rankings, hi-score, increasing difficulty.

INVADERS £7.95
48 marching invaders drop bombs that slowly erode your defences, and two types of spaceship (normal and double speed) fly over releasing large bombs that penetrate through your defences. Increasing difficulty, hi-score, rankings, superb graphics and sound.

WORLD GEOGRAPHY £7.95
This program covers 166 countries which are divided into 8 categories of difficulty. Each country is pinpointed on an accurate hi-resolution screen map of the world, and the user is asked the capital and/or population. At the end of the test, the percentage of correct answers is given, so that the student can easily monitor his increasing geographical knowledge.

FRUIT MACHINE £7.95
Probably the best fruit machine implementation on the market. This program has it all . . . HOLD, NUDGE, GAMBLE, spinning reels, realistic fruits and sound effects, multiple winning lines. This is THE fruit machine program to buy.

CONSTELLATION £7.95
This fascinating program enables the user to "view the stars" from any point on the Earth's surface, on any date and at any time. A total of 455 stars in 50 constellations may be viewed, and the "telescope" may be moved up, down, left or right, zoomed in or zoomed out. The stars can be displayed by magnitude or constellation.

DISASSEMBLER £7.95
A relocatable disassembler which, unlike some similar programs, allows the disassembled source code to be output to memory. It may then be modified and re-assembled. Other features: page-mode option, output to printer if required, output of ASCII symbols if required.

WE PAY UP TO 20% ROYALTIES FOR HIGH QUALITY BBC MICRO AND ELECTRON PROGRAMS

SUPERIOR SOFTWARE LTD.
Dept. EU4, Regent House,
Skinner Lane, Leeds 7
Tel: 0532 459453

OUR GUARANTEE
(1) All our software is available before we advertise.
(2) All our software is despatched within 48 hours by first-class post.
(3) In the unlikely event that any of our software fails to load, return your cassette to us and we will immediately send a replacement.

PROGRAMPOWERMICROPOWER PROGRAMPOWERMICROPOWER PROGRAMPOWERMICROPOWER PROGRAMPOWERMICROPOWER

KILLER GORILLA

A SUPERB B.B.C. MICRO AND ELECTRON PROGRAM FROM BRITAIN'S LEADING SOFTWARE HOUSE!

Scale the ironwork tower to answer the maiden's cries for help. Race along girders, career along conveyors, climb ladders and jump onto moving elevators. Leap the barrels and fireballs or smash them with the hammer. A sensational machine code game for the BBC micro and the Electron

Only **£7.95** (inc.VAT)

AVAILABLE FROM ALL GOOD RETAILERS INCLUDING SELECTED BRANCHES OF W. H. SMITH, JOHN MENZIES AND BOOTS.

WRITTEN ANY PROGRAMS? WE PAY 20% ROYALTIES!

The following top titles are available for both the BBC Micro and Electron:- **Positron £6.95 /Bandits at 3 o'Clock £6.95/ Moonraider £7.95/Croaker £7.95/Felix in the Factory £7.95/ Felix and the Fruit Monsters £7.95/Chess £7.95/Draw £9.95/ Escape from Moonbase Alpha £7.95/Cybertron Mission £7.95/ Swoop £7.95/Intergalactic Trader £8.95/**

SHOWROOM: NORTHWOOD HOUSE, NORTH STREET, LEEDS LS7 2AA Tel: (0532) 458800

MAIL ORDER: 8/8a REGENT STREET, LEEDS LS7 4PE, Tel: (0532) 683186/696343 P & P: 55p per order

SPECIAL OFFER: Deduct £1 per cassette when ordering two or more

PROGRAMPOWERMICROPOWER PROGRAMPOWERMICROPOWER PROGRAMPOWERMICROPOWER PROGRAMPOWERMICROPOWER